

**PEMBANGUNAN APLIKASI PERANGKAT BERGERAK
PROGRAM KB BERBASIS ANDROID DENGAN MENERAPKAN
*USER CENTERED DESIGN***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Lia Rosalina
NIM: 135150201111018



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

PEMBANGUNAN APLIKASI PERANGKAT BERGERAK PROGRAM KB BERBASIS ANDROID
DENGAN MENERAPKAN USER CENTERED DESIGN

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Lia Rosalina

NIM: 135150201111018

Skripsi ini telah diuji dan dinyatakan lulus pada
1 Agustus 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Adam Hendra Brata, S.Kom., M.T., M.Sc.

NIK: 201607 900105 1 001

Dosen Pembimbing II



Lutfi Fanani, S.Kom., M.T., M.Sc.

NIK: 201607 890217 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D

NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 6 Agustus 2018



Lia Rosalina

NIM: 135150201111018

KATA PENGANTAR

Dengan menyebut nama Allah SWT yang Maha Pengasih Lagi Maha Penyayang, yang telah melimpahkan rahmat, hidayah, dan inayah-Nya kepada penulis, sehingga penulis dapat menyelesaikan tugas akhir dengan judul “Pembangunan Aplikasi Perangkat Bergerak Program KB Berbasis Android Dengan Menerapkan User Centered Design”.

Dalam penyelesaian tugas akhir ini, penulis juga mendapat banyak sekali bantuan, masukan, dukungan moral dan materil. Maka penulis mengucapkan banyak terima kasih sebesar-besarnya kepada yang terhormat:

1. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
2. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Bapak Agus Wahyu Widodo, S.T, M.Cs selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
4. Bapak Adam Hendra Brata, S.Kom., M.T., M.Sc. dan Bapak Lutfi Fanani S.Kom., M.T., M.Sc. selaku Dosen Pembimbing yang telah banyak membantu dan mengarahkan dalam proses penulisan skripsi ini.
5. Seluruh dosen Fakultas Ilmu Komputer Universitas Brawijaya yang telah meberikan ilmu kepada penulis.
6. Kedua orang tua penulis, Ayah Sunardi dan Mama Luluk Setiyo, adik Putri Pamungkas, serta seluruh keluarga penulis yang selalu memberikan dukungan, nasihat, serta doa yang tidak putus selama masa penelitian berlangsung.
7. Kepada Mas Adam yang selalu memberikan dukungan semangat dan dengan sabar membimbing penulis hingga penulis dapat menyelesaikan tugas akhir ini.
8. Kerabat dekat penulis yaitu Ahmad Nur Sofyan yang telah memberikan dukungan semangat tanpa henti.
9. Teman – teman penulis yaitu Diah Shinta Dewi, Arie Krisnoanto, Rusjda Widya, Artiyana Prasetya yang selalu ada di sisi penulis dalam keadaan suka maupun duka.
10. Teman – teman penulis yaitu Amalia Kartika S.Kom, Sifasani Q Fauzia S.Kom, Kukuh Wiliam S.Kom, yang telah banyak membantu dalam pembuatan tugas akhir penulis.
11. Kerabat penulis Nuril Charisma, Ibu Nur Saidah, serta teman-teman bidan yang bersedia membantu dalam melakukan pengujian pada aplikasi KBku.

12. SABEB Entertainment yang telah memberikan pengalaman baru dalam pertemanan, memberikan pengalaman untuk lebih produktif dengan membuat konten di *youtube*.

13. Keluarga BIOS yang telah memberikan pengalaman baru dalam berorganisasi.

Penulis menyadari bahwa dalam proses penulisan skripsi ini masih terdapat kesalahan dan kekurangan di beberapa bagian. Karena itu kritikan dan saran yang membangun sangat penulis harapkan demi perbaikan dan pengembangan skripsi ini dimasa mendatang. Semoga skripsi ini bermanfaat bagi siapapun yang membaca dan menerapkannya dalam kehidupan sehari-hari.

Malang, 6 Agustus 2018

Penulis

liaroosalina@gmail.com



ABSTRAK

Program Keluarga Berencana merupakan program pemerintah yang digunakan untuk memenuhi kebutuhan kesehatan reproduksi yang berkualitas bagi masyarakat. Dengan program KB maka dapat menurunkan tingkat kematian ibu dan bayi serta menekan tingkat kehamilan pada wanita guna membangun keluarga kecil berkualitas. Dari banyaknya faktor yang dapat mempengaruhi kegagalan program KB, faktor sumber daya manusia merupakan faktor utama, karena banyak wanita usia subur yang belum mengetahui bahwa program KB harus dilaksanakan tepat waktu. Faktor lupa dan ketidakmauan untuk bertanya kepada bidan membuat celah bagi peneliti untuk mengembangkan sebuah aplikasi program KB berbasis android. Data dari statista.com menjelaskan bahwa terdapat 75% pengguna *smartphone* dengan sistem operasi Android hingga akhir tahun 2017. Berdasarkan hal tersebut maka dibuatlah aplikasi program KB dengan menerapkan metode *User Centered Design* yang dilakukan dengan melihat dari pengalaman pengguna sehingga dapat membuat aplikasi yang memiliki nilai *usability*.

Berdasarkan dari hasil pengujian *usability* dengan menggunakan metode *task scenario* dan menggunakan metode *use quissionaire*. Didapatkan hasil sebesar 100% untuk metode *task scenario* dan hasil sebesar 96,20% untuk metode *use quissonaire*. Hal ini menunjukkan bahwa tingkat keberhasilan aplikasi KBku memasuki kategori sangat baik.

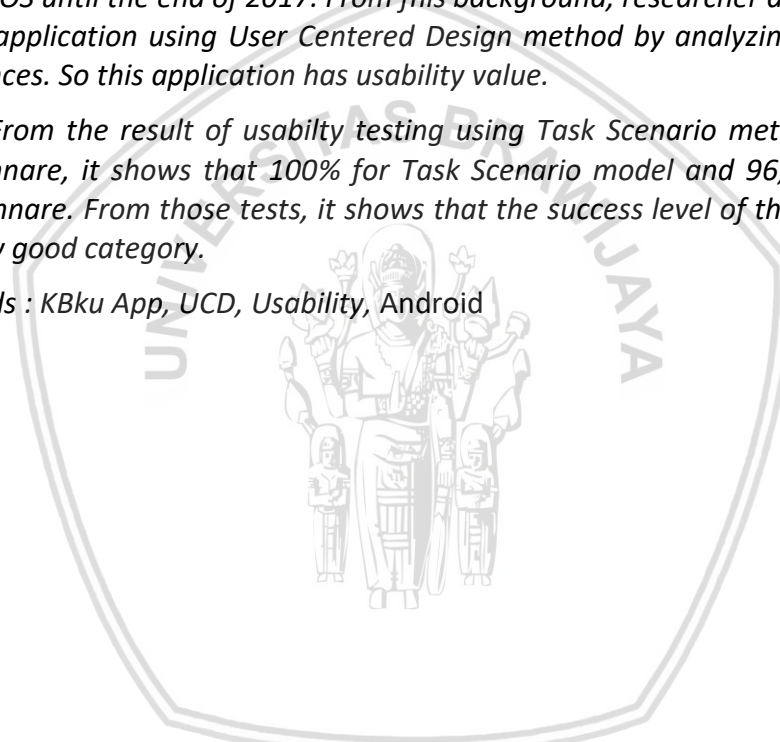
Kata kunci: Aplikasi KBku, UCD, *Usability*, Android

ABSTRACT

Birth control program is a government program to fulfill the quality of reproductive health needs for society. By planning birth control, it can reduce the amount of death of mothers and children as well as minimize the amount of pregnancy to make high quality of small family. From many factors which affect the failure of birth control, the main factor is human resources itself, because many women who are in fertile age did not know that birth control should be done on time. Forget and unwillingness to ask the nurse about birth control become the background for researcher to develop application of birth control program using android. Data from statista.com shows that there are 75% smartphone users with android OS until the end of 2017. From this background, researcher develops birth control application using User Centered Design method by analyzing from users experiences. So this application has usability value.

From the result of usability testing using Task Scenario method and Use questionnaire, it shows that 100% for Task Scenario model and 96,20% for Use Questionnaire. From those tests, it shows that the success level of the application is in very good category.

Keywords : KBku App, UCD, Usability, Android



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xvi
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah	3
1.3 Tujuan	4
1.4 Manfaat.....	4
1.5 Batasan masalah	4
1.6 Sistematika pembahasan.....	5
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Kajian Pustaka	6
2.2 Program Keluarga Berencana	6
2.2.2 Kontrasepsi.....	7
2.3 Android	9
2.4 User Centered Design (UCD).....	9
2.5 <i>K-Nearest Neighbor</i>	12
2.6 <i>Firestore</i>	12
2.7 <i>Google Maps Service</i>	13
2.8 <i>Prototyping</i>	13
2.9 Pengujian Perangkat Lunak.....	14
2.9.1 Pengujian Unit.....	14
2.9.2 Pengujian Validasi	14
2.9.3 Pengujian <i>Usability</i>	14

2.9.4 Pengujian <i>Compatibility</i>	18
BAB 3 METODOLOGI	19
3.1 Studi Literatur	19
3.2 Analisis Kebutuhan	20
3.3 Perancangan Sistem.....	21
3.4 Evaluasi Pengguna	22
3.5 Implementasi Sistem	22
3.6 Pengujian Sistem.....	22
3.7 Pengambilan Kesimpulan dan Saran	23
BAB 4 REKAYASA KEBUTUHAN.....	24
4.1 Analisis Kebutuhan	24
4.1.1 Gambaran Umum Aplikasi	24
4.1.2 Analisis Kebutuhan Iterasi 1.....	39
4.1.3 Analisis Kebutuhan Iterasi 2.....	43
BAB 5 PERANCANGAN DAN IMPEMENTASI.....	45
5.1 Perancangan Sistem.....	45
5.1.1 Perancangan Arsitektur Sistem.....	45
5.1.2 Perancangan Basis Data	46
5.1.3 Perancangan <i>Sequence Diagram</i>	49
5.1.4 Perancangan <i>Class Diagram</i>	51
5.1.5 Perancangan Antarmuka.....	55
5.2 Perancangan Sistem Iterasi 1.....	63
5.2.1 Perancangan Basis Data Iterasi 1	63
5.2.2 Perancangan <i>Class Diagram Iterasi 1</i>	66
5.2.3 Perancangan Antarmuka Iterasi 1.....	68
5.3 Perancangan Sistem Iterasi 2.....	71
5.3.1 Perancangan <i>Class Diagram Iterasi 2</i>	71
5.3.2 Perancangan Antarmuka Iterasi 2.....	71
5.3.3 Perancangan Algoritme.....	72
5.3.4 Perancangan Antarmuka <i>Screenflow</i>	75
5.4 Implementasi Sistem	78
5.4.1 Spesifikasi Sistem	78

5.4.2 Batasan Implementasi.....	79
5.4.3 Implementasi Basis Data	79
5.4.4 Implementasi <i>Class</i>	81
5.4.5 Implementasi Kode Program	82
5.4.6 Implementasi Antarmuka	86
BAB 6 PENGUJIAN	95
6.1 Pengujian Fungsional	95
6.1.1 Pengujian Unit.....	95
6.1.2 Pengujian Validasi	101
6.2 Pengujian Non-Fungsional	110
6.2.1 Pengujian <i>Usability</i>	110
6.2.2 Pengujian <i>Compatibility</i>	115
6.2.3 Analisis Hasil Pengujian Unit	116
6.2.4 Analisis Hasil Pengujian Validasi	117
6.2.5 Analisis Hasil Pengujian <i>Usability</i>	117
6.2.6 Analisis Hasil Pengujian <i>Compatibility</i>	119
BAB 7 PENUTUP	120
7.1 Kesimpulan.....	120
7.2 Saran	121
DAFTAR PUSTAKA.....	122
LAMPIRAN A WAWANCARA TAHAP PERTAMA.....	126
LAMPIRAN B PERHITUNGAN KNN REKOMENDASI KB	133

DAFTAR TABEL

Tabel 2.1 Tabel Kuisiner <i>USE</i>	17
Tabel 2.2 Tabel Nilai.....	17
Tabel 2.3 Tabel <i>Skala Linkert</i>	18
Tabel 4.1 Identifikasi Aktor	25
Tabel 4.2 Identifikasi Kebutuhan Fungsional Pengguna KB	26
Tabel 4.3 Identifikasi Kebutuhan Fungsional Pengguna KB	26
Tabel 4.4 Identifikasi Kebutuhan Fungsional Akseptor.....	27
Tabel 4.5 Identifikasi Kebutuhan Fungsional Akseptor.....	28
Tabel 4.6 <i>Use Case Scenario</i> Pengguna KB Daftar Akun	29
Tabel 4.7 <i>Use Case Scenario</i> Pengguna KB Masuk ke Sistem	30
Tabel 4.8 <i>Use Case Scenario</i> Akseptor Melihat Informasi KB	31
Tabel 4.9 <i>Use Case Scenario</i> Akseptor Melihat Riwayat KB.....	32
Tabel 4.10 <i>Use Case Scenario</i> Akseptor Membuat Riwayat KB	32
Tabel 4.11 <i>Use Case Scenario</i> Akseptor menghapus Riwayat KB	33
Tabel 4.12 <i>Use Case Scenario</i> Akseptor Melihat Riwayat Kalender KB	34
Tabel 4.13 <i>Use Case Scenario</i> Akseptor Membuat Kalender KB.....	35
Tabel 4.14 <i>Use Case Scenario</i> Akseptor Menerima Pemberitahuan KB	35
Tabel 4.15 <i>Use Case Scenario</i> Akseptor Melihat Pemberitahuan KB	36
Tabel 4.16 <i>Use Case Scenario</i> Akseptor Mencari Rumah sakit.....	37
Tabel 4.17 <i>Use Case Scenario</i> Akseptor Melihat Profil	37
Tabel 4.18 <i>Use Case Scenario</i> Akseptor Mengubah Profil	38
Tabel 4.19 <i>Use Case Scenario</i> Akseptor Keluar Dari Sistem	39
Tabel 4.20 Identifikasi Kebutuhan Fungsional Pengguna KB	40
Tabel 4.21 <i>Use Case Scenario</i> Akseptor Membuat Rekomendasi KB	41
Tabel 4.22 <i>Use Case Scenario</i> Akseptor Melihat Rekomendasi KB.....	42
Tabel 4.23 <i>Use Case Scenario</i> Akseptor Membuat Kalender KB.....	43
Tabel 5.1 Keterangan Skema Basis Data Riwayat KB	47
Tabel 5.2 Keterangan Skema Basis Data Pemberitahuan KB.....	48
Tabel 5.3 Keterangan Skema Basis Data Akseptor	49
Tabel 5.4 Keterangan Skema Basis Data Pemberitahuan KB Iterasi 1	64

Tabel 5.5 Keterangan Skema Basis Data Rekomendasi KB	65
Tabel 5.6 Spesifikasi Perangkat Keras	78
Tabel 5.7 Spesifikasi Perangkat Lunak	78
Tabel 5.8 Spesifikasi Perangkat Bergerak	79
Tabel 5.9 Implementasi <i>Class</i>	81
Tabel 5.10 Penjelasan <i>Method</i> simpan().....	83
Tabel 5.11 Penjelasan <i>Method</i> buatNotifikasi().....	84
Tabel 5.12 Penjelasan <i>Method</i> buatNotifikasi().....	85
Tabel 6.1 Kasus Uji Pengujian Unit <i>Method</i> simpan	97
Tabel 6.2 Kasus Uji Pengujian Unit <i>Method</i> buatNotifikasi	98
Tabel 6.3 Kasus Uji Pengujian Unit <i>Method</i> rekomendasi.....	100
Tabel 6.4 Kasus Uji Masuk ke Sistem	101
Tabel 6.5 Kasus Uji Daftar Akun	102
Tabel 6.6 Kasus uji Melihat Informasi KB.....	102
Tabel 6.7 Kasus Uji Membuat Rekomendasi KB	102
Tabel 6.8 Kasus Uji Lihat Rekomendasi KB.....	103
Tabel 6.9 Kasus Uji Lihat Riwayat Kalender KB	103
Tabel 6.10 Kasus Uji Buat Kalender KB	103
Tabel 6.11 Kasus Uji Menerima Pemberitahuan KB	104
Tabel 6.12 Kasus Uji Lihat Pemberitahuan KB	104
Tabel 6.13 Kasus Uji Mencari Rumah Sakit Terdekat.....	104
Tabel 6.14 Kasus Uji Melihat Profil	105
Tabel 6.15 Kasus Uji Mengubah Profil	105
Tabel 6.16 Kasus Uji Keluar dari Sistem	105
Tabel 6.17 hasil pengujian validasi sisi pengguna.....	106
Tabel 6.18 Pengujian <i>Usability</i> Menggunakan <i>Task Scenario</i>	110
Tabel 6.19 Hasil Pengujian <i>Task Scenario</i>	112
Tabel 6.20 Pengujian <i>Usability</i> Menggunakan <i>USE Quissionaire</i>	113
Tabel 6.21 Hasil Pengujian <i>Compatibility</i>	115
Tabel 6.22 Analisis Hasil Pengujian <i>Task Scenario</i>	117
Tabel 6.23 Analisis Hasil Pengujian <i>USE Quissionaire</i>	117

DAFTAR GAMBAR

Gambar 1.1 Infografik Pengguna <i>Smartphone</i>	2
Gambar 2.1 Proses <i>User-Centered Design</i>	11
Gambar 3.1 Gambar Alur Metodologi Penelitian	19
Gambar 4.1 Gambar Deskripsi Aplikasi.....	24
Gambar 4.2 Gambar <i>Use Case Diagram</i>	28
Gambar 4.3 Gambar <i>Use Case Diagram</i> Iterasi 1	41
Gambar 4.4 Gambar <i>Use Case Diagram</i> Iterasi 2	44
Gambar 5.1 Arsitektur Sistem Aplikasi <i>Mobile</i>	45
Gambar 5.2 Arsitektur Sistem Aplikasi KBku	46
Gambar 5.3 Gambar Perancangan Basis Data Riwayat KB	47
Gambar 5.4 Gambar Perancangan Basis Data Pemberitahuan KB.....	48
Gambar 5.5 Gambar Perancangan Basis Data Akseptor.....	48
Gambar 5.6 Gambar <i>Sequence Diagram</i> Akseptor Melihat Informasi KB.....	49
Gambar 5.7 Gambar <i>Sequence Diagram</i> Akseptor Melihat Informasi KB.....	50
Gambar 5.8 Gambar <i>Sequence Diagram</i> Akseptor Pemberitahuan KB.....	50
Gambar 5.9 Gambar Perancangan <i>Class Diagram</i> Aplikasi KBku	52
Gambar 5.10 Gambar Perancangan <i>Class Diagram</i> Paket Kalender KB	52
Gambar 5.11 Gambar Perancangan <i>Class Diagram</i> Paket Akseptor	53
Gambar 5.12 Gambar Perancangan <i>Class Diagram</i> Paket Riwayat KB.....	53
Gambar 5.13 Gambar Perancangan <i>Class Diagram</i> Paket Rumah Sakit.....	54
Gambar 5.14 Gambar Perancangan <i>Class Diagram</i> Paket Informasi KB	54
Gambar 5.15 Gambar Perancangan <i>Class Diagram</i> Paket Profil	55
Gambar 5.16 Gambar Perancangan <i>Class Diagram</i> Beranda	55
Gambar 5.17 Gambar Perancangan Antarmuka Halaman Masuk ke Sistem	56
Gambar 5.18 Gambar Perancangan Antarmuka Halaman Daftar Akun	56
Gambar 5.19 Gambar Perancangan Antarmuka Halaman Beranda.....	57
Gambar 5.20 Gambar Perancangan Antarmuka Halaman Daftar Informasi KB... 58	
Gambar 5.21 Gambar Perancangan Antarmuka Halaman Detail Informasi KB ... 58	
Gambar 5.22 Gambar Perancangan Antarmuka Halaman Daftar Riwayat KB 59	
Gambar 5.23 Gambar Perancangan Antarmuka Halaman Buat Riwayat KB	59

Gambar 5.24 Gambar Perancangan Antarmuka Halaman Detail Riwayat KB	60
Gambar 5.25 Gambar Perancangan Antarmuka Halaman Daftar Kalender	60
Gambar 5.26 Gambar Perancangan Antarmuka Halaman Buat Kalender KB	61
Gambar 5.27 Gambar Perancangan Antarmuka Halaman Isi Pemberitahuan KB	61
Gambar 5.28 Gambar Perancangan Antarmuka Halaman Mencari Rumah sakit	62
Gambar 5.29 Gambar Perancangan Antarmuka Halaman Profil	62
Gambar 5.30 Gambar Perancangan Antarmuka Halaman Edit Profil.....	63
Gambar 5.31 Gambar Perancangan Basis Data Pemberitahuan KB Iterasi 1	64
Gambar 5.32 Gambar Perancangan Basis Data Rekomendasi KB	65
Gambar 5.33 Gambar Perancangan <i>Class Diagram</i> Aplikasi KBku Iterasi 1	66
Gambar 5.34 Gambar Perancangan <i>Class Diagram</i> Paket Kalender KB Iterasi 1 .	67
Gambar 5.35 Gambar Perancangan <i>Class Diagram</i> Paket Akseptor	68
Gambar 5.36 Gambar Perancangan Antarmuka Halaman Rekomendasi KB	69
Gambar 5.37 Gambar Perancangan Antarmuka Halaman Hasil Rekomendasi KB	69
Gambar 5.38 Gambar Perancangan Antarmuka Halaman Riwayat Kalender KB .	70
Gambar 5.39 Gambar Perancangan Antarmuka Halaman Buat Kalender KB	70
Gambar 5.40 Gambar Perancangan <i>Class Diagram</i> Aplikasi KBku Iterasi 2	71
Gambar 5.41 Gambar Perancangan Antarmuka Halaman Isi Pemberitahuan KB Iterasi 2.....	72
Gambar 5.42 Gambar Perancangan Algoritme Simpan Kalender KB	73
Gambar 5.43 Gambar Perancangan Algoritme Buat Notifikasi	74
Gambar 5.44 Gambar Perancangan Algoritme Mencari Rumah sakit.....	75
Gambar 5.45 Gambar Perancangan Antarmuka <i>Screenflow</i>	77
Gambar 5.46 Implementasi Basis Data Akseptor	79
Gambar 5.47 Implementasi Basis Pemberitahuan KB	80
Gambar 5.48 Implementasi Struktural Basis Data Rekomendasi KB	80
Gambar 5.49 Implementasi Basis Data Rekomendasi KB	81
Gambar 5.50 Implementasi <i>Method</i> simpan().....	83
Gambar 5.51 Implementasi <i>Method</i> buatNotifikasi().....	84
Gambar 5.52 Implementasi <i>Method</i> rekomendasi()	85
Gambar 5.53 Implementasi Masuk Ke Sistem	86
Gambar 5.54 Implementasi Daftar Akun	87

Gambar 5.55 Implementasi Halaman Beranda.....	87
Gambar 5.56 Implementasi Daftar Informasi KB.....	88
Gambar 5.57 Implementasi Informasi KB.....	89
Gambar 5.58 Implementasi Rekomendasi KB.....	89
Gambar 5.59 Implementasi Hasil Rekomendasi KB.....	90
Gambar 5.60 Implementasi Kalender KB.....	91
Gambar 5.61 Implementasi Daftar Riwayat Kalender KB.....	91
Gambar 5.62 Implementasi Pemberitahuan KB.....	92
Gambar 5.63 Implementasi Rumah Sakit.....	93
Gambar 5.64 Implementasi Halaman Ubah Profil.....	93
Gambar 5.65 Implementasi Halaman Profil.....	94
Gambar 6.1 Kode Program <i>Method simpan</i>	96
Gambar 6.2 <i>Flowgraph method simpan</i>	96
Gambar 6.3 Kode Program <i>Method buatNotifikasi</i>	97
Gambar 6.4 <i>Flowgraph Method buatNotifikasi</i>	98
Gambar 6.5 Kode Program <i>Method rekomendasi</i>	99
Gambar 6.6 <i>Flowgraph Method rekomendasi</i>	100

DAFTAR LAMPIRAN

LAMPIRAN A WAWANCARA TAHAP PERTAMA.....	126
A.1 Daftar Persona.....	126
A.2 Form Wawancara Terhadap Pengguna Program KB	128
A.3 Form Wawancara Evaluasi Kebutuhan Iterasi 1.....	131
A.4 Form Wawancara Evaluasi Kebutuhan Iterasi 2.....	132
LAMPIRAN B PERHITUNGAN KNN REKOMENDASI KB	133
B.1 Proses Perhitungan Algoritme K-NN pada Rekomendasi KB.....	133



BAB 1 PENDAHULUAN

1.1 Latar belakang

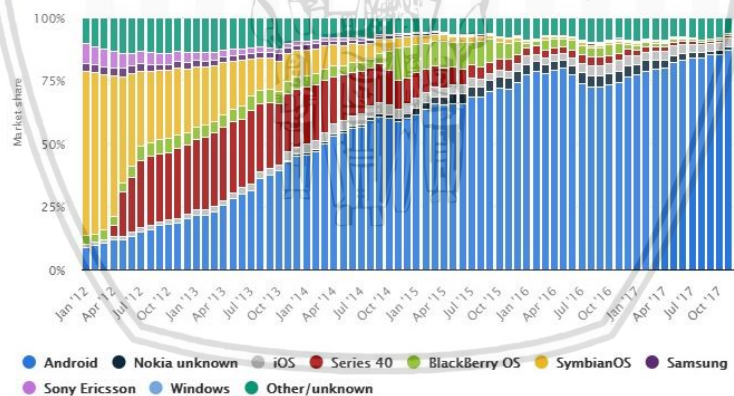
Sejak tahun 1970 program Keluarga Berencana (KB) digunakan oleh pemerintah yang bertujuan untuk memenuhi kesehatan reproduksi yang berkualitas bagi masyarakat, menurunkan tingkat kematian ibu dan bayi serta menanggulangi masalah kesehatan reproduksi dalam rangka membangun keluarga kecil berkualitas. Kontrasepsi adalah sebuah alat yang digunakan untuk menekan tingkat kehamilan pada wanita berusia subur, kontrasepsi sendiri memiliki berbagai macam tipe, ada yang bersifat permanen dan sementara. Kontrasepsi memiliki beberapa metode, metode yang pertama adalah Metode Kontrasepsi Jangka Panjang (MKJP), pada metode MKJP terdiri dari IUD, implant, MOW, dan MOP. Metode yang kedua adalah metode Kontrasepsi Non-MKJP, pada metode Non-MKJP terdiri dari pil, suntik, dan kondom (Trisanti & Nasriyah, 2016). Dalam pemilihan program KB juga ada beberapa faktor external yang harus diperhatikan, yaitu konsekuensi dari kegagalan atau kehamilan yang tidak diinginkan, besarnya keluarga yang diinginkan seperti keinginan memiliki beberapa orang anak, serta faktor-faktor norma budaya, lingkungan, agama, pasangan, serta orang tua (Affandi, et al., 2012).

Selain masalah tingginya Laju Pertumbuhan Penduduk dan tingginya angka kelahiran total, Indonesia juga masih menghadapi masalah Angka Kematian Ibu (AKI). Berdasarkan SDKI 2012 Angka Kematian Ibu justru meningkat yaitu 359 per 100.000 kelahiran hidup. Salah satu upaya yang dapat dilakukan dalam untuk mengatasi masalah tersebut diatas adalah dengan penguatan pelayanan Program Keluarga Berencana melalui penggunaan kontrasepsi. United Nation Family Planning Association (UNFPA), menyebutkan bahwa satu dari tiga kematian yang berhubungan dengan kehamilan atau melahirkan bisa dihindari jika semua wanita memiliki akses terhadap layanan kontrasepsi. Dalam mengatasi masalah pertumbuhan penduduk yang tinggi, pemerintah telah menerapkan kebijakan penggunaan kontrasepsi yang rasional, efektif dan efisien diantaranya adalah penggunaan Metode Kontrasepsi Jangka Panjang (MKJP). Menurut WHO dan American College of Obstetricians dan Gynecologist (ACOG) bahwa Metode Kontrasepsi Jangka Panjang merupakan metode kontrasepsi yang paling efektif. Berdasarkan hasil mini survey BKKBN pada akseptor KB aktif, mengungkapkan bahwa cukup banyak akseptor KB yang menggunakan cara KB dengan tidak rasional atau tidak sesuai dengan umur ibu, jumlah anak yang diinginkan dan kondisi kesehatan ibu (Nuryati & Fitria, 2014).

Menurut wawancara yang peneliti lakukan kepada Bidan Subaedah A.Md.Keb, faktor yang mempengaruhi tingkat kegagalan program KB adalah sumber daya manusia yang kurang, faktor ekonomi, faktor agama, faktor ketidakcocokan alat kontrasepsi yang berhubungan dengan hormon, dan yang terakhir adalah ketika mengikuti program keluarga berencana tetapi tidak sesuai dengan tanggal kembali. Dari kelima faktor tersebut, yang berpengaruh terhadap

kegagalan program KB adalah faktor sumber daya manusia yang kurang, karena banyak sekali wanita yang tidak begitu memahami tentang program keluarga berencana, seperti program apa yang cocok dengan kondisi keluarga dan hormon tubuhnya, serta informasi apa saja yang sangat penting terkait dengan program keluarga berencana yang sedang diikuti. Selain itu banyak pengguna KB yang tidak tepat waktu untuk kembali melakukan program KB, karena pada dasarnya program KB berhasil jika pengguna KB melakukan program tersebut secara rutin tanpa adanya keterlambatan. Dan untuk saat ini, penyimpanan tanggal untuk melakukan KB selanjutnya dan juga penyimpanan data terakhir seperti tekanan darah dan berat badan pada saat melakukan KB sebelumnya, hanya disimpan didalam selembar kertas kecil, tidak jarang kertas tersebut hilang dan rusak. Tidak sedikit pengguna KB yang harus kembali lagi ke rumah sakit atau bidan tempat langganannya hanya untuk meminta kartu KB.

Berdasarkan permasalahan yang telah didapatkan diatas maka penulis merasa diperlukannya pembuatan sebuah aplikasi yang dapat dijalankan pada *smartphone*. Pada saat ini *smartphone* bukan hal asing lagi bagi masyarakat luas. Perkembangan *smartphone* sendiri terus bertambah setiap tahunnya. Pada Gambar 1.1 menjelaskan bahwa perkembangan *smartphone* setiap tahunnya terus meningkat, khususnya bagi *smartphone* dengan *operating system* Android. Pada bulan Oktober 2017 pengguna *smartphone* dengan *operating system* Android mencapai jumlah 75%. Jumlah 75% tersebut didapatkan dari situs web Statista: The Statistic Portal (Statista, 2017).



Gambar 1.1 Infografik Pengguna Smartphone

Sumber: Statista, 2017

Dari permasalahan yang sudah dijelaskan sebelumnya dan melihat pertambahan pengguna dari *smartphone* yang terus berkembang setiap tahunnya, maka diperlukan untuk membuat sebuah aplikasi perangkat bergerak yang digunakan untuk mencatat tanggal-tanggal penting yang digunakan untuk melakukan program KB dan juga memberikan rekomendasi KB yang sesuai dengan apa yang diperlukan oleh pengguna KB. Pada aplikasi perangkat bergerak ini, berisi tentang informasi-informasi program KB, menyimpan jadwal dan tanggal penting yang digunakan untuk melakukan proses KB, rekomendasi KB apa yang sesuai dengan kebutuhan pengguna KB dari segi ekonomi, serta letak rumah sakit

terdekat. Untuk mempermudah pengguna KB dalam menggunakan aplikasi KB nantinya, maka penulis menggunakan metode pendekatan *User Centered Design* (UCD) dalam melakukan proses pembuatan aplikasi. UCD merupakan strategi yang digunakan untuk melakukan desain dengan menggunakan pengalaman pengguna untuk membuat suatu produk dan membuat suatu teknologi informasi (LeRouge & Wickramasighe, 2013). Untuk melakukan proses rekomendasi KB agar pengguna KB mendapatkan sebuah rekomendasi yang cocok dengan kebutuhannya, dari berbagai macam metode yang digunakan untuk mengambil keputusan penulis menggunakan *K-Nearest Neighbor* (KNN) untuk menyelesaikan masalah rekomendasi KB, karna KNN lebih efektif dan menghasilkan data yang lebih akurat pada jumlah data *training* yang besar, karena itu algoritme KNN sangat cocok untuk diterapkan pada penelitian ini. *K – Nearest Neighbour* merupakan salah satu metode yang digunakan dalam pengklasifikasian. Prinsip kerja *K – Nearest Neighbour* adalah mencari jarak terdekat antara data yang akan dievaluasi dengan K tetangga (*neighbor*) terdekatnya dalam data *training*. Parameter yang digunakan untuk melakukan rekomendasi adalah umur istri, pendidikan istri, pendidikan suami, jumlah anak, agama istri, istri sedang bekerja, pekerjaan suami, standar hidup, dan pandangan masyarakat. Selanjutnya dari sembilan kategori diatas akan diidentifikasi menjadi tiga kelas yaitu tidak menggunakan, jangka panjang dan jangka pendek (Nugroho, et al., 2015). Setelah aplikasi dibuat, aplikasi akan diuji menggunakan pengujian *usability* dan *compatibility*. Pengujian *usability* menurut *International Organization or Standariditazion* (1998) digunakan untuk melihat apakah aplikasi yang dibuat sudah memiliki efektifitas, efesien, dan memperoleh kepuasan dalam konteks penggunaannya. Sedangkan pengujian *compatibility* digunakan untuk menguji apakah aplikasi dapat berjalan baik diberbagai perangkat *mobile* dikarenakan perangkat *mobile* pengguna KB memiliki bermacam-macam spesifikasinya seperti versi OS Android, ukuran layar perangkat *mobile* dan resolusi layarnya (Pusuluri, 2006) .

Berdasarkan latar belakang yang sudah dijelaskan diatas maka dilakukan sebuah penelitian yang berjudul **“Pembangunan Aplikasi Perangkat Bergerak Program KB Berbasis Android Dengan Menerapkan User Centered Design”**. Sehingga hasil dari dilakukannya penelitian ini dapat menghasilkan sebuah aplikasi android yang bernama “KBku” dengan antarmuka yang mudah digunakan oleh pengguna, serta dapat memberikan pemberitahuan waktu program KB yang digunakan oleh pengguna KB yang akan atau sedang melakukan program keluarga berencana untuk menekan tingkat kegagalan program keluarga berencana.

1.2 Rumusan masalah

Berdasarkan uraian latar belakang diatas dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana hasil analisis kebutuhan dan perancangan pada aplikasi KBku dengan menerapkan metode *user centered design*?

2. Bagaimana hasil implementasi pada aplikasi KBku dengan menerapkan metode *user centered design*?
3. Bagaimana hasil dari pengujian pada aplikasi KBku?
4. Apakah aplikasi KBku dapat dengan mudah digunakan oleh pengguna?

1.3 Tujuan

Penelitian ini dilakukan untuk bertujuan:

1. Melakukan analisis kebutuhan dengan menggunakan metode *user centered design* agar pengguna KB merasa nyaman menggunakan aplikasi KBku.
2. Melakukan perancangan dan implementasi dengan menggunakan metode *user centered design* sehingga hasil dari perancangan dan implementasi benar-benar sesuai dengan kebutuhan pengguna KB.
3. Mendapatkan hasil pengujian yang sangat baik, dengan harapan aplikasi KBku dapat mudah digunakan dan dapat membantu dalam melakukan program KB.

1.4 Manfaat

Hasil dari penelitian ini diharapkan memberikan manfaat untuk berbagai pihak, khususnya wanita pada usia subur yang sedang atau akan melakukan program KB, agar dapat memiliki pengetahuan yang baik mengenai alat kontrasepsi dan program keluarga berencana, sehingga dapat mengurangi tingkat kegagalan dan ketidaknyamanan penggunaan alat kontrasepsi.

1.5 Batasan masalah

Agar permasalahan pada penelitian ini lebih terfokus dan tidak terjadi pelebaran topik, maka penelitian ini di batasi pada hal:

1. Aplikasi ini terbatas pada pengolahan yang terkait dengan informasi, pemberitahuan tempat rumah sakit terdekat, serta pemberitahuan KB yang memberitahukan waktu untuk melakukan KB.
2. Pemberitahuan KB waktu KB hanya terbatas pada metode KB jangka pendek, yaitu suntik satu bulan dan tiga bulan.
3. Metode pendekatan yang digunakan dalam membangun aplikasi perangkat bergerak pemberitahuan KB waktu program KB menggunakan *user centered design*.
4. Data diperoleh dari hasil wawancara dan observasi dengan calon pengguna sistem dan bidan setempat.
5. Parameter yang digunakan untuk menentukan hasil dari rekomendasi menggunakan parameter dari jurnal yang dilakukan pada penelitian sebelumnya.
6. Aplikasi ini di bangun dengan menggunakan bahasa java android.
7. *Tools* yang di gunakan adalah android studio.
8. Basis data yang digunakan adalah *firebase*.

1.6 Sistematika pembahasan

Secara garis besar sistematika pembahasan dari laporan penelitian ini terdiri dari tujuh bab, sebagai berikut:

BAB 1 PENDAHULUAN

Pada bab satu, yaitu bab pendahuluan berisi tentang latar belakang kenapa perlu di buatnya suatu penelitian tentang kegagalan KB serta kenapa perlu di buat aplikasi catatan program KB, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan proposal.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab dua, yaitu bab landasan kepastakaan berisi tentang pengertian dan konsep serta teori-teori yang di gunakan sebagai dasar penunjang penyusunan dari penelitian.

BAB 3 METODOLOGI PENELITIAN

Pada bab tiga, yaitu bab metodologi penelitian berisi tentang metode yang di gunakan dalam melakukan penelitian, yaitu studi literatur, analisis, perancangan, implementasi, hingga pengujian dari aplikasi yang akan dibuat.

BAB 4 REKAYASA KEBUTUHAN

Pada bab empat, yaitu bab rekayasa kebutuhan berisi tentang analisis kebutuhan yang terdapat pada sistem, kebutuhan di sini meliputi kebutuhan fungsional maupun non fungsional. Hasil dari proses analisis tersebut akan dibuat sebagai acuan dalam melakukan perancangan pada sistem.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab lima, yaitu bab perancangan dan implementasi berisi tentang perancangan dan implementasi aplikasi sesuai dengan hasil dari rekayasa kebutuhan dalam membangun aplikasi program KB.

BAB 6 PENGUJIAN

Pada bab enam, yaitu bab pengujian berisi tentang teknik pengujian terhadap sistem yang sudah dibangun. Hasil dari pengujian tersebut akan digunakan sebagai dasar dalam melakukan analisis dari kehandalan pada sebuah aplikasi dan apakah aplikasi tersebut sudah sesuai dengan apa yang di butuhkan.

BAB 7 PENUTUP

Pada bab terakhir, yaitu bab tujuh penutup berisi tentang kesimpulan dan saran yang berasal dari hasil penelitian berdasarkan uraian-uraian dari bab-bab sebelumnya yang nantinya di gunakan untuk pengembangan lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

Landasan pustaka pada penelitian yang penulis lakukan membahas tentang penelitian yang telah dilakukan sebelumnya dan literatur yang di gunakan untuk mendukung keberhasilan dari penelitian yang dilakukan oleh penulis.

2.1 Kajian Pustaka

Berdasarkan penelitian yang sudah dilakukan sebelumnya dengan judul Sistem Pendukung Keputusan Untuk Pemilihan Metode Kontrasepsi Pada Pasangan Usia Subur Dengan Algoritma *K-Nearest Neighbor* (KNN) mendapatkan kesimpulan bahwa dengan menerapkan K-NN untuk menghitung dan memberikan hasil dari rekomendasi KB untuk memilih metode kontrasepsi mendapatka tingkat akurasi perhitungan sebesar 95% (Nugroho, et al., 2015). Berbeda dengan sistem yang telah dibuat oleh Nugroho, aplikasi KBku berfokus kepada pembuatan apikasi yang menerapkan metode *user centered design* yang berfokus kepada pengguna aplikasi agar pengguna aplikasi dapat mendapatkan kemudahan serta kenyamanan dalam menggunakan apkikasi KBku. Dan juga memberikan fitur tambahan lain yang digunakan untuk memberitahukan pemberitahuan waktu untuk melakukan program KB.

2.2 Program Keluarga Berencana

Keluarga berencana adalah usaha untuk mengatur angka kelahiran, kematian serta meningkatkan kesejahteraan masyarakat dan memberikan peran kepada masyarakat terkait dengan pendewasaan usia perkawinan. Hal tersebut telah di atur dalam Undang-Undang Republik Indonesia Nomor 52 Tahun 2009 yang membahas tentang perkembangan kependudukan dan pembangunan keluarga, dalam Undang-undang tersebut, terdapat bab IV yang menjelaskan tentang perkembangan kependudukan, dan paragram kedua menjelaskan tentang keluarga berencana yang di jelaskan pada pasal 20 sampai dengan pasal 29. Menurut penjelasan pasal 20 sampai dengan pasal 29 maka dapat di ambil kesimpulan bahwa program keluarga berencana dilakukan untuk membantu calon dan pasangan suami istri mengambil keputusan terkait dengan hak reproduksi secara bertanggung jawab (Soleha, 2016).

Tujuan Keluarga berencana adalah (Wilopo, 1997):

1. Pasangan atau individu bisa menentukan jumlah dan jarak anak yang akan dimiliki.
2. Memperoleh informasi dan jaminan untuk merealisasikan keinginannya.
3. Wajib memperoleh informasi terkait dengan pilihan metode KB yang ada.
4. Menyediakan metode-metode yang efektif dan aman untuk pasangan serta individu yang sedang atau ingin melakukan program KB.

Berdasarkan tujuan keluarga berencana diatas maka aplikasi program KBku memiliki sebuah fitur informasi KB yang digunakan sebagai tujuan keluarga berencana.

2.2.2 Kontrasepsi

Menurut Kamus Besar Bahasa Indonesia (KBBI) kontrasepsi adalah cara yang digunakan untuk melakukan suatu pencegahan kehamilan dengan menggunakan alat atau obat. Dalam melakukan program kontrasepsi terdapat berbagai macam metode, berikut adalah macam metode yang dilakukan dalam melakukan program keluarga berencana (KKB, 2011). Berbagai macam metode yang dijelaskan berikut digunakan sebagai isi dari informasi KB pada aplikasi KBku.

a. Metode Amonera Laktasi (MAL)

MAL adalah kontrasepsi yang dilakukan menggunakan pemberian air susu ibu secara eksklusif dan tidak ada tambahan makanan.

MAL digunakan sebagai alat kontrasepsi jika:

1. Ibu sedang dalam tahap menyusui secara penuh, dan jumlah pemberian ASI adalah lebih dari delapan kali sehari.
2. Belum datang masa haid
3. Umur bayi yang disusui kurang dari enam bulan

Efektifnya KB MAL digunakan selama enam bulan saja, lebih dari itu disarankan menggunakan metode lainnya.

b. Metode Keluarga Berencana Alamiah (KBA)

Metode KBA ini juga dikenal sebagai Metode Ovulasi Bilings atau MOB. Jika wanita pada usia subur melakukan program ini maka wanita tersebut harus mulai memberi tanda dan mulai mencari tahu kapan masa suburnya berlangsung. Metode ini sangat efektif jika dilakukan secara disiplin, dan kelebihan metode ini adalah tidak memiliki efek samping.

c. Senggama Terputus

Senggama terputus adalah merupakan sebuah metode kontrasepsi tradisional. Manfaat kontrasepsi ini yang pertama adalah sangat efektif jika dilakukan dengan baik dan benar, yang kedua tidak mengganggu produksi ASI, yang ketiga dapat digunakan sebagai pendukung metode dari program KB yang lainnya, yang keempat tidak ada efek samping jika melakukan program KB ini, kelima program KB ini dapat digunakan setiap waktu, dan yang terakhir adalah tidak membutuhkan biaya dalam jika menggunakan program KB ini.

d. Metode Barrier

Metode Barrier adalah metode program keluarga berencana dengan cara menghalangi masuknya sel sperma kedalam sel telur yang bersifat sementara. Dalam metode Barrier ada beberapa cara yang bisa di gunakan yaitu dengan menggunakan kondom, diafragma, spermisida.

e. Kontrasepsi Kombinasi (Hormon Estrogen dan Progesteron)

Kontrasepsi kombinasi merupakan kontrasepsi yang dilakukan dengan cara menyuntikan hormon sintesis progesterone dan esterogen.

Dalam kontrasepsi kombinasi terdapat dua cara yaitu menggunakan pil kombinasi dan menggunakan suntik kombinasi.

Berikut adalah keterangan dari masing-masing kontrasepsi kombinasi:

1. Pil Kombinasi

Pil kombinasi sangat efektif dan *reversible*, diminum setiap hari, pada bulan pertama efek sampingnya adalah mual dan pendarahan bercak, dapat digunakan oleh semua ibu dalam usia reproduksi, dan sangat tidak dianjurkan ketika ibu dalam masa menyusui.

Jenis dari pil kombinasi adalah:

- a. Monofasik: pil dalam kemasan 21 tablet dan mengandung hormon aktif esterogen atau progestin dalam dosis yang sama dan terdiri dari 7 tablet tanpa hormon aktif.
- b. Bifasik: pil dalam kemasan 21 tablet dan mengandung hormon aktif esterogen atau progestin dalam dua dosis yang berbeda, dan terdiri dari 7 tablet tanpa hormon aktif.
- c. Trifasik: pil dalam kemasan 21 tablet yang mengandung hormon aktif estrogen atau progestin dengan tiga dosis yang berbeda, dan terdiri dari 7 tablet tanpa hormon aktif.

2. Suntikan Kombinasi

Suntikan kombinasi digunakan untuk menekan ovulasi. Keuntungan menggunakan suntikan kombinasi adalah resiko dari suntikan ini tidak mengganggu kesehatan, tidak diperlukan pemeriksaan dalam, bisa digunakan dalam jangka panjang.

f. Kontrasepsi Progestin

Kontrasepsi progestin merupakan kontrasepsi dengan cara disuntik dalam jangka waktu tiga bulan sekali, dalam suntikan progestin ini hanya mengandung hormone sintesis progesterone. Alat ini sangat efektif digunakan karna pemakaiannya yang tiga bulan sekali dengan cara disuntik, kontrasepsi ini aman untuk digunakan, dapat digunakan oleh semua wanita dalam usia subur, sangat cocok untuk masa laktasi karena tidak menekan produksi ASI.

g. Alat Kontrasepsi Dalam Rahim (AKDR)

Alat kontrasepsi dalam Rahim adalah suatu alat yang digunakan untuk mencegah kehamilan secara efektif, reversible yang terbuat dari plastic atau logam kecil yang dimasukkan kedalam uterus melalui kanalis servikalis. AKDR ini merupakan program kontrasepsi jangka panjang dapat digunakan sampai 10 tahun. Dapat digunakan oleh seluruh wanita dalam usia subur, namun tidak boleh digunakan oleh perempuan yang terpapa infeksi menular seksual. Jenis dari AKDR adalah:

1. AKDR CuT-380A

2. NOVA T
3. CuT-380Aq

h. Kontrasepsi Mantap

Pada metode ini jarang sekali ditemukan efek samping dari jangka panjang maupun jangka pendek. Namun resiko menggunakan kontrasepsi mantap pun juga terjadi. Resikonya terjadi ketika melakukan proses pembedahan dan anastesi.

1. Tubektomi

Tubektomi adalah metode kontrasepsi yang digunakan untuk perempuan yang sudah tidak menginginkan anak lagi. Pada kontrasepsi ini diperlukan pembedahan, sebelum dilakukan pembedahan maka perlu pemeriksaan fisik dan pemeriksaan tambahan lainnya. Ini digunakan untuk memastikan apakah klien tersebut sudah sesuai untuk melakukan metode tubektomi.

2. Vasektomi

Vasektomi adalah metode kontrasepsi yang digunakan untuk pria yang tidak menginginkan anak lagi. Pada vasektomi diperlukan prosedur bedah sehingga diperlukan pemeriksaan fisik dan pemeriksaan lainnya untuk memastikan apakah klien tersebut sudah sesuai untuk melakukan metode vasektomi

2.3 Android

Android adalah sebuah *open source* yang didesain untuk aplikasi perangkat bergerak. Platform ini di usung oleh *Google* dan dimiliki oleh *Open Handset Alliance*. Tujuan aliansi ini adalah untuk inovasi pada bidang perangkat bergerak dan menawarkan konsumen beberapa kelebihan, yaitu tidak mahal dan pengalaman yang baik dalam menggunakan perangkat bergerak. Android merupakan sebuah *platform* terbuka yang memisahkan antara perangkat keras dan lunak yang berjalan diatas perangkat keras (Burnette, 2009). Untuk implementasi sistem program KB, maka digunakan perangkat bergerak berbasis Android.

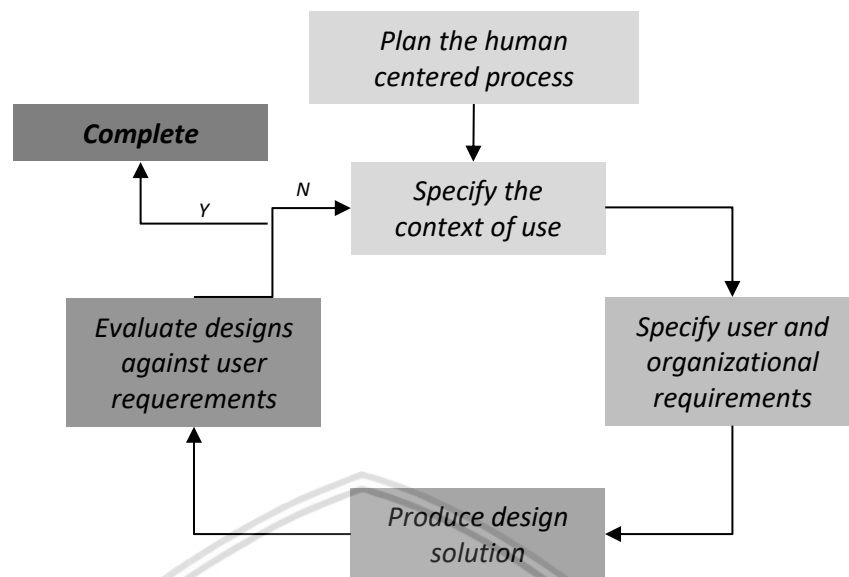
2.4 User Centered Design (UCD)

User Centered Design (UCD) adalah sebuah istilah yang digunakan untuk menggambarkan proses dari sebuah desain yang mempengaruhi bagaimana desain tersebut terbentuk. Istilah dari UCD sendiri berasal dari Donald Norman Laboratorium riset University of California San Diego (UCSD) pada tahun 1980an dan kemudian menjadi banyak digunakan oleh peneliti lainnya setelah Donald Norman menerbitkan buku *User Centered System Design: New Perspectives on Human-Computer Interaction* pada tahun 1986. Norman mengakui kebutuhan dan kepentingan pengguna berfokus pada desain. Norman menjelaskan bahwa ada empat saran dasar bagaimana desain itu seharusnya:

1. Membuatnya mudah untuk ditentukan tindakan apa yang mungkin harus dilakukan.
2. Membuat hal-hal yang terlihat, termasuk model konseptual dari sistem, alternatif tindakan, dan hasil dari tindakan.
3. Memudahkan untuk mengevaluasi kondisi sistem pada saat ini.
4. Mengikuti pemetaan alami antara niat dan tindakan yang diperlukan, antara tindakan dan efek yang dihasilkan, serta diantara informasi yang terlihat dan interpretasi dari sistem itu sendiri.

Berdasarkan saran rekomendasi diatas peran dari desain adalah untuk memfasilitasi tugas untuk pengguna dan untuk memastikan bahwa pengguna dapat menggunakan produk dengan usaha seminimal mungkin untuk belajar bagaimana menggunakannya (Abrams, et al., 2004). Menurut ISO 13407 merupakan standar international yang didirikan pada tahun 1999, standar ini menjelaskan tentang metode *user centered design*, yang digunakan untuk mengelola proses desain. ISO 13407 menggambarkan proses desain yang berpusat pada pengguna dari empat aspek yang berbeda, empat aspek tersebut adalah alasan UCD, Prinsip-prinsip UCD, Perencanaan UCD, dan kegiatan UCD (Jokela, et al., 2003).

1. Alasan UCD adalah dengan adanya UCD maka dapat menjelaskan manfaat bahwa sistem dapat digunakan untuk meningkatkan kemudahan pengguna, karena kepuasan pengguna merupakan focus utama metode ini.
2. Prinsip-prinsip UCD memiliki empat prinsip umum yang cirinya adalah berpusat pada pengguna, empat prinsip tersebut adalah pertama keterlibatan aktif pengguna yang jelas faham tentang kebutuhan dari pengguna serta tugas-tugas yang dilakukan pengguna, kedua sesuai dengan alokasi fungsi antara pengguna serta teknologi yang digunakan, ketiga iterasi dan desain solusi, keempat adalah perancangan interaktif, Sistem yang akan dikembangkan harus didefinisikan, dirancang, dilakukan test berkali-kali. Berdasarkan test yang telah dilakukan secara berulang kali tersebut maka akan ditentukan fungsi, antarmuka, sistem bantuan, dokumentasi pengguna, serta pendekatan pelatihannya .
3. Perencanaan UCD merupakan desain seluruh proses pembangunan sistem. berikut merupakan ulustrasi dari perencanaan UCD.



Gambar 2.1 Proses User-Centered Design

Sumber : (ISO 13407:1999)

Gambar 2.1 menjelaskan sebuah proses UCD yang berdasarkan ISO 13407:1999. Berikut adalah keterangan terkait Gambar 2.1.

1. *Plan the human centered process* merupakan proses awal untuk mengidentifikasi pengguna yang sesuai dengan sistem yang akan dibuat. Pada proses pembuatan aplikasi KBKU ini melakukan identifikasi pengguna, pengguna yang sesuai dengan aplikasi KBKU adalah wanita usia subur yang sedang dalam proses akan melakukan atau sudah menggunakan program KB.
2. *Specify the context of use* merupakan proses untuk memahami karakteristik dan harapan pengguna, pekerjaan, serta lingkungan tempat pengguna akan menggunakan sistem, dan menentukan kebutuhan sistem minimal dan optimal. Untuk karakteristik pengguna aplikasi KBKU adalah wanita usia subur yang tidak mengetahui tentang KB apa yang sesuai dengan kondisi keadaan keluarga dan keinginan untuk memiliki anak.
3. *Specify the user and organizational requirements* merupakan proses untuk melakukan perluasan terhadap aktifitas kebutuhan fungsional sistem dengan membuat pernyataan yang eksplisit. Untuk mengetahui apa yang dibutuhkan oleh pengguna aplikasi KBKU, penulis melakukan wawancara terhadap wanita usia subur serta bidan.
4. *Produce design solutions* merupakan proses untuk menggunakan pengetahuan yang sudah ada untuk melakukan pengembangan dari solusi perancangan, maka membuat solusi perancangan lebih konkrit contohnya dengan menggunakan simulasi atau *prototype*. Sebelum

melakukan pembuatan aplikasi KBKU, penulis melakukan desain awal dengan menggambar di kertas kemudian di implementasikan dengan cara melakukan desain di aplikasi *coreldraw* kemudian dijalankan di aplikasi *marvel*.

5. *Evaluate designs against requirements* merupakan proses yang digunakan oleh pengembang akan melakukan evaluasi terhadap desain yang dilakukan pada tahapan sebelumnya. Tahap ini digunakan untuk mengetahui apakah kebutuhan dari pengguna telah tercapai.

2.5 K-Nearest Neighbor

Tujuan dari algoritma *K-Nearest Neighbor* (K-NN) adalah mengklasifikasikan objek baru berdasarkan atribut dan data latih. KNN memiliki beberapa kelebihan yaitu ketika data latih yang memiliki banyak *noise* algoritma ini akan menjadi efektif ketika data training bersifat besar. Sementara kelemahan KNN adalah nilai dari parameter *k* (jumlah tetangga terdekat) harus diinisialisasi terlebih dahulu. Selain itu, proses KNN hanya bergantung pada jarak dari satu data ke data lainnya, sehingga tidak ada referensi atribut / fitur mana yang berpengaruh untuk mendapatkan hasil terbaik (Gerhana, et al., p. 2017).

Klasifikasi algoritma KNN menurut (Gerhana, et al., p. 2017) memiliki beberapa langkah sebagai berikut:

- a. Tentukan parameter *k*.
- b. Hitung jarak antara data yang akan dievaluasi dengan seluruh data latih. Jarak antara data yang akan dievaluasi dengan seluruh data latih dapat menggunakan rumus pada Persamaan 2.1.

$$d(x, y) = \sum_{i=1}^N \sqrt{(x_i - y_i)^2} \quad (2.1)$$

Dimana x_i , y_i adalah nilai dari atribut data ke- i dan n adalah jumlah keseluruhan data latih (Panwar et al, 2016).

- c. Urutkan jarak yang telah diperoleh (mulai dari jarak terpendek).
- d. Tentukan jarak terpendek dengan urutan *k*.
- e. Padukan kelas yang sesuai.
- f. Temukan jumlah kelas dari tetangga terdekat dan atur kelas sebagai kelas data yang akan dievaluasi.
- g. Klasifikasi data.

2.6 Firebase

Firebase merupakan sebuah teknologi yang dapat digunakan untuk membuat sebuah aplikasi tanpa melakukan pemrograman dari sisi *server*, hal ini membuat proses pembangunannya menjadi lebih mudah dan cepat. Pada *firebase* kita dapat melakukan kontrol terhadap aplikasi, dan menggunakan data tanpa perlu khawatir bagaimana data akan disimpan, karna sifat *firebase* yang *realtime* maka data akan secara otomatis disimpan dan disinkronkan secara otomatis. (Kumar, et al., 2016). Saat API diimplementasikan pada aplikasi *cross-platform*, *Firebase* dapat

memberikan fitur dengan bagis kode sederhana. Fitur-fitur yang ditawarkan Firebase yaitu Autentifikasi, *Hosting*, *Messaging*, *Analytics*, *Storage*, *Real-time Database*, *Crash Reporting*, *App Indexing* dan *Admob*. Berdasarkan fitur- fitur tersebut berikut adalah fitur yang diimplementasikan dalam basis data aplikasi program KB:

1. *Authentication*

Authentication ini merupakan salah satu layanan yang diberikan oleh Firebase untuk memberikan izin hanya kepada pengguna tertentu yang sudah diberikan izin untuk mengakses aplikasi. Autentikasi ini menyediakan fitur *login* dan *register* melalui *email*, nomor ponsel, anonim hingga mendukung beberapa media sosial seperti *Gmail*, *Github*, *Twitter*, *Facebook* untuk mempermudah pengguna menggunakan aplikasi.

2. *Real-time Database*

Firebase real-time Database adalah database yang diposisikan pada cloud. Datanya disimpan dalam bentuk JSON dan disikronkan secara real-time tiap kali pengguna aplikasi terhubung. Ketika pengembang membuat aplikasi lintas platform dengan SDK Android, iOS, dan JavaScript, semua klien akan berbagi sebuah *instance Realtime Database* dan menerima *update* data terbaru secara otomatis.

2.7 Google Maps Service

Google memiliki fasilitas *google maps service* menggunakan *google play service library*, yang digunakan untuk pengguna dengan fungsi menunjukan setiap lokasi pada peta, atau melihat rute yang ada pada peta. Google menyediakan fasilitas menggunakan *google play service library* yang dapat di *download* dari para pengembang di android SDK manager. *Google Maps API* diperlukan untuk melakukan intregasi peta dalam aplikasi android. *Google maps* dapat diintegrasikan dengan melakukan langkah-langkah berikut (Doshi, et al., 2014):

1. *Install* dan konfigurasi *google play service library* dari android SDK manager.
2. Menambahkan *google play service* ke *library* dari sebuah aplikasi.
3. Mengambil *android certificate* yaitu SHA1 *fingerprint* yang digunakan sebagai kunti untuk melakukan *sign in* ke dalam aplikasi.
4. Membuat sebuah proyek *console google API*.
5. Mendapatkan key *google maps API*.
6. Menghasilkan API key *unique*, yang ditambahkan ke aplikasi.

Berdasarkan hal diatas *google maps service* digunakan untuk mencari rumah sakit terdekat pada aplikasi KBku.

2.8 Prototyping

Prototype merupakan model dari aspek sesungguhnya yang akan dikembangkan, model ini bersifat *representative* dari produk akhirnya. Dalam

melakukan pengembangan suatu sistem terkadang mengalami keraguan mengenai efektifitas, efisiensi dan kualitas algoritme yang sedang dikembangkan. Model *prototype* dilakukan untuk menyambungkan ketidak pahaman calon *user* mengenai hal teknis dan digunakan untuk memperjelas spesifikasi kebutuhan yang diinginkan oleh calon *user* (Kinanti, et al., 2016).

2.9 Pengujian Perangkat Lunak

Sebelum diberikan kepada pelanggan, perangkat lunak harus melakukan tahapan pengujian. Hal ini digunakan agar perangkat lunak yang dihasilkan bisa lebih terjamin. Berikut adalah pengujian yang dilakukan oleh peneliti yakni:

2.9.1 Pengujian Unit

Pengujian unit atau biasa disebut *white box testing* dilakukan dengan mengecek seluruh jalur dalam alur kerja program sehingga dapat dideteksi kesalahan yang mungkin terjadi (Pressman, 2010).

2.9.2 Pengujian Validasi

Pengujian validasi atau biasa disebut *black box testing* dilakukan dengan melakukan testing terhadap aplikasi dan hasilnya akan dibandingkan dengan kebutuhan yang telah di analisis sebelumnya (Pressman, 2010).

2.9.3 Pengujian Usability

Menurut *International Organization for Standardization* (1998) menjabarkan bahwa *usability* digunakan untuk mencapai tujuan dengan efektifitas, efisien, dan memperoleh kepuasan dalam konteks penggunaannya.

2.9.3.1 Ukuran Tingkat Usability

Ukuran tingkat *usability* adalah mencakup 10 hal berikut (Harrison, et al., 2013):

1. Efektifitas
Hal ini digunakan sebagai performansi pengguna ketika menggunakan sistem yang digunakan.
2. Efisiensi
Meliputi kebutuhan sumber daya, yang mencakup daya, usaha, waktu, dan biaya yang digunakan untuk mencapai pemakaian sistem.
3. *Learnability*
Digunakan untuk tingkat kemudahan sistem ketika dipelajari, diukur melalui waktu yang diperlukan untuk mempelajari penggunaan sistem selama beberapa waktu.
4. *Memorability*
Cara pemakaian sistem setelah pengguna tidak berinteraksi dengan sistem tersebut selama beberapa waktu.

5. *Fleksibilitas*
Berkaitan dengan variasi pengerjaan suatu *taks* sistem.
6. *Error*
Kesalahan yang tinggi pada saat penggunaan sistem mengindikasikan rendahnya *usability* sistem yang bersangkutan.
7. *Utilitas*
Acuan tingkat fungsionalitas sebuah sistem yang digunakan pengguna untuk menyelesaikan tugas.
8. *Safety*
Keamanan yang digunak untuk mencegah kerusakan fatal pada sistem dari kondisi yang tidak diinginkan.
9. *Satisfaction*
Kepuasan pengguna terhadap sistem yang digunakannya.
10. *Attitude*
Mengukur kriteria perilaku sistem dari bagaimana pengguna menerima dan merasa puas terhadap sistem yang digunakannya.

2.9.3.2 Langkah Pengujian *Usability*

Terdapat langkah-langkah untuk melakukan pengujian *usability*, berikut merupakan penjelasan langkah untuk melakukan pengujian *usability* (Nurhadryani, et al., 2011):

1. Menyiapkan sketsa, *prototype* yang ingin dilakukan pengujian.
2. Merancang alat ukur.
Alat ukur disesuaikan dengan komponen yaitu efektivitas, efisiensi dan kebutuhan pengguna. Pengukuran nilai efektivitas dan efisiensi menggunakan lembar observasi, sedangkan untuk penilaian kepuasan pengguna menggunakan *post-task* kuisioner.
3. Melaksanakan pengujian
Pada saat observasi, responden melakukan langkah-langkah sesuai dengan elemen observasi.
4. Menganalisis Hasil
Tingkat efektivitas, efisiensi, *learnability*, *satisfaction* diukur menggunakan tingkat keberhasilan pengguna dalam menyelesaikan elemen observasi dengan perhitungan sebagai berikut:

$$\text{Efektivitas (\%)} = \frac{\sum_{i=1}^n x_i}{n} \times 100\% \quad (2.2)$$

$$\text{Efisiensi (\%)} = \frac{\sum_{i=1}^n x_i}{n} \times 100\% \quad (2.3)$$

$$\text{Learnability (\%)} = \frac{\sum_{i=1}^n x_i}{n} \times 100\% \quad (2.4)$$

$$\text{Satisfaction (\%)} = \frac{\sum_{i=1}^n x_i}{n} \times 100\% \quad (2.5)$$

Dimana x_i adalah nilai keberhasilan responden ke- i , $x_i = \{0,1\}$.

Sedangkan kepuasan pengguna didapat dari *persentase* perbandingan antara nilai kepuasan dengan perkalian bobot maksimum *skala Likert* (5) dengan jumlah responden (n).

Dimana x_i adalah nilai keberhasilan responden ke- i , $x_i = \{0,1,2,3,4,5\}$.

Sedangkan nilai *usability* sendiri didapat dari rata-rata efektivitas, efisiensi dan kepuasan pengguna.

$$\text{Usability (\%)} = \frac{(\text{efektivitas} + \text{efisiensi} + \text{learnability} + \text{satisfaction})}{3} \times 100\% \quad (2.6)$$

2.9.3.3 Jumlah Partisipan Pengujian *Usability*

Untuk melakukan sebuah pengujian *usability* dengan hasil terbaik maka dibutuhkan setidaknya lima orang partisipan tidak peduli dengan perbedaan *platform* (Nielsen, 2012). Dengan menggunakan lima orang partisipan, hasil yang didapatkan mendekati rasio maksimum pengujian pengguna. Pada aplikasi KBku, terdapat delapan partisipan.

2.9.3.4 *Task Scenario*

Task Scenario digunakan untuk menggambarkan cerita dan konteks mengapa pengguna menggunakan sebuah aplikasi atau sebuah *web*. Dengan menggunakan metode ini hasil dan pertanyaan dicapai untuk menentukan bagaimana pengguna dapat menggunakan sebuah aplikasi dan *web*. *Task scenario* sangat baik untuk digunakan merancang antarmuka dan melakukan pengujian. Dari *task scenario* ini dapat mencapai tujuan untuk mendefinisikan bagaimana pengguna dapat mencapai tujuannya pada aplikasi atau *web*, dengan mengidentifikasi berbagai kemungkinan dan hambatan potensial. Terdapat tiga macam tipe dari *task scenario*, berikut merupakan penjelasan dari tipe *task scenario* (Gov, 2018).

1. *Goal or task based scenario*

Hanya menyatakan apa yang akan dilakukan oleh pengguna. Tanpa menyertakan informasi bagaimana pengguna akan menyelesaikan *scenario*.

2. *Elaborated scenarios*

Skenario yang diuraikan dengan memberi banyak lebih banyak rincian erita ke pengguna.

3. Full scale task scenarios

Mencangkup langkah-langkah untuk menyelesaikan tugas. Dengan menggunakan *full task scenarios* dapat dilaporkan semua langkah yang dibutuhkan pengguna untuk menyelesaikan tugas.

Pada penelitian ini menggunakan *task scenario* jenis *full scale task scenarios*. Hal ini dikarenakan agar peneliti dapat mengetahui bahwa pengguna KB dapat menggunakan seluruh fitur dengan baik.

2.9.3.5 USE Quesionnaire

Pada pengujian *usability* menggunakan kuestioner *USE (useful, satisfaction, and easy of use)* untuk mengukur *learnability, efficiency, memorability, errors dan satisfaction* dengan menggunakan skala likert untuk perhitungan hasilnya. Format kuestioner untuk melakukan pengujian *usability* dapat dilihat pada Tabel 2.1 dengan *skala Likert* sebagai acuan untuk menghitung hasil kuestioner dapat dilihat pada Tabel 2.2.

Tabel 2.1 Tabel Kuisisioner USE

USEFULNESS		STS	TS	N	S	SS
1						
2						
EASE OF USE						
3						
4						
EASE OF LEARNING						
5						
6						
SATISFACTION						
7						
8						

Tabel 2.2 Tabel Nilai

STS	TS	N	S	SS
1	2	3	4	5

Keterangan :

STS = Sangat tidak setuju

TS = Tidak Setuju

N = Netral

S = Setuju

SS = Sangat setuju

Tabel 2.3 Tabel *Skala Linkert*

No	Interval	Nilai
1	0% - 19.99%	Sangat Buruk
2	20% - 39.99%	Buruk
3	40% - 59.99%	Cukup
4	60% - 79.99%	Baik
5	80% - 100%	Sangat Baik

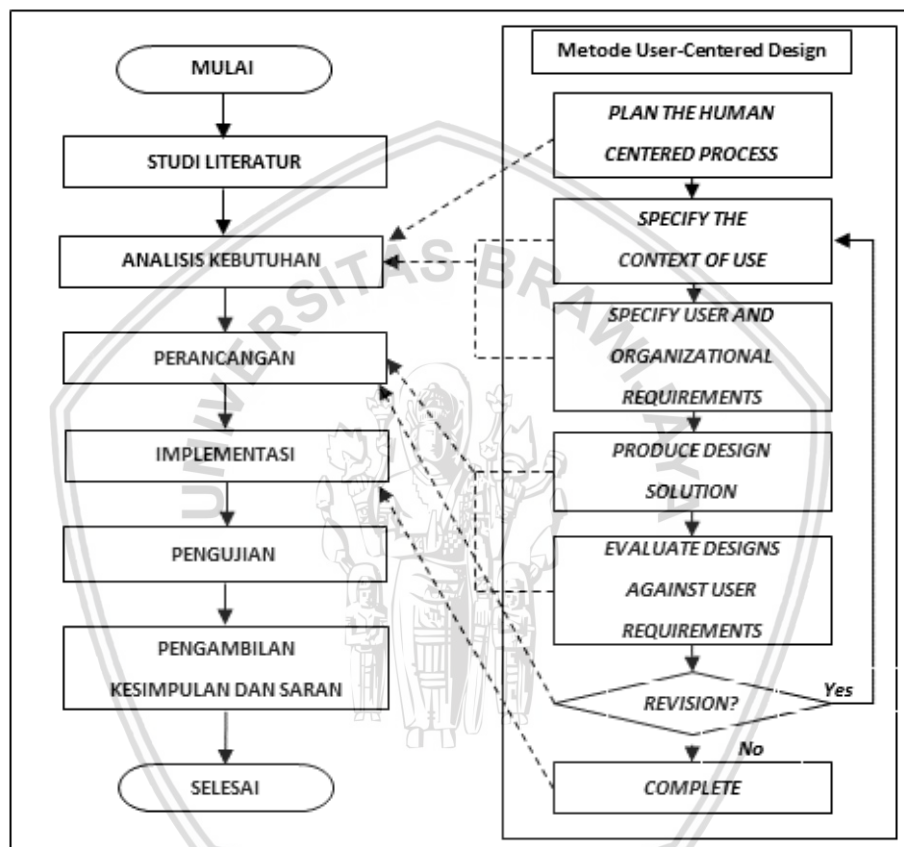
Responden atau sasaran kuesioner tersebut adalah pengguna KB sebagai pengguna utama aplikasi KBku serta beberapa badan, kemudian setelah mengisi kuesioner akan dilakukan perhitungan dari hasil kuisisioner tersebut (Rahadi, 2014).

2.9.4 Pengujian *Compatibility*

Pengujian *compatibility* merupakan pengujian yang digunakan untuk mengetahui apakah aplikasi yang dikembangkan mampu berjalan pada berbagai macam *device platform* yang berbeda. Android merupakan sebuah proyek *open source* maka semua pabrik perangkat keras dapat membangun perangkat yang dapat menjalankan sistem operasi android. Namun perangkat android dapat dikatakan *compatible* jika dapat dengan benar menjalankan aplikasi yang ditulis untuk lingkungan eksekusi android. Pengembang aplikasi perlu mempertimbangkan apakah aplikasi yang dibuat sudah kompatibel dengan perangkat android, hal ini dikarenakan beberapa fitur tidak terdapat pada semua perangkat. Untuk melakukan pengujian, dapat dilakukan berdasarkan karakteristik perangkat yaitu perangkat fitur, *platform* versi, dan layar konfigurasi (android, 2018).

BAB 3 METODOLOGI

Pada bab ini peneliti membahas metode yang digunakan untuk membangun aplikasi KBKU, metode yang digunakan adalah penerapan dari UCD. Diantaranya adalah studi literatur, analisis kebutuhan, perancangan sistem, perancangan antarmuka, evaluasi pengguna, implementasi sistem, pengujian sistem, dan pengambilan kesimpulan serta saran.



Gambar 3.1 Gambar Alur Metodologi Penelitian

3.1 Studi Literatur

Studi literatur merupakan sebuah metode yang digunakan untuk mendapatkan dasar teori sebagai sumber acuan bagi peneliti untuk melakukan suatu penelitian dalam pengembangan aplikasi. Teori yang berkaitan dengan penelitian ini adalah:

1. Kajian Pustaka
2. Program Keluarga Berencana
3. Android
4. *User Centered Design*
5. *K- Nearest Neighbor*

6. *Firebase*
7. *Google Maps Service*
8. *Prototyping*
9. Pengujian Unit
10. Pengujian Validasi
11. Pengujian *Usability*
12. Pengujian *Compatibility*

3.2 Analisis Kebutuhan

Pada subbab ini menjelaskan tentang analisis kebutuhan yang diperlukan dalam pembuatan aplikasi KBKU. Untuk analisis kebutuhan pada penelitian ini penulis merujuk kepada metode perencanaan UCD (*User Centered Design*) yaitu pada tahapan *specify the context of use* dan *specify the user and organizational requirements*. Pada proses *specify the context of use* penulis melakukan proses pengumpulan data dengan memahami karakteristik, harapan, pekerjaan, serta lingkungan tempat pengguna yang akan menggunakan sistem dan menentukan kebutuhan sistem minimal dan optimal, untuk pengguna aplikasi KBKU sendiri merupakan wanita usia subur yang sedang atau akan menggunakan program KB. Tahapan *specify the context of use* terdapat pada bab empat dengan subbab analisa kebutuhan, gambaran umum sistem, dan lingkungan sistem. Untuk *specify the user and organizational requirements* merupakan proses yang dilakukan untuk mendapatkan kebutuhan fungsional yang sesuai dengan apa yang pengguna butuhkan. Untuk mengetahui kebutuhan calon pengguna aplikasi KBKU penulis melakukan wawancara secara langsung kepada bidan yang menangani proses keluarga berencana dan juga kepada wanita yang sedang melakukan program KB, tidak hanya dengan bidan peneliti juga melakukan wawancara langsung dengan wanita usia subur yang menggunakan program KB. Untuk total calon pengguna aplikasi yang diwawancarai adalah berjumlah delapan orang, yang terdiri dari tiga bidan dan lima pengguna KB. Tahapan *specify the user and organizational requirements* sendiri terdapat pada bab empat dengan subbab analisis kebutuhan fungsional, hasil dari kebutuhan fungsional tersebut digambarkan dalam *use case diagram* dan di implementasikan kedalam *use case scenario*.

Setelah kedua proses UCD tersebut mendapatkan hasil maka dilakukan proses pengumpulan data, data yang sudah didapat akan dianalisis, kemudian hasil dari analisis kebutuhan akan dijabarkan dengan menggambarkan gambaran umum sistem, dan dijadikan kebutuhan fungsional dan non fungsional. Kebutuhan fungsional berisi tentang fitur-fitur yang dibutuhkan oleh pengguna dan disediakan oleh sistem. Sedangkan kebutuhan non fungsional adalah kebutuhan yang terkait dengan kualitas dan batasan sistem tersebut diaplikasikan. Kebutuhan non fungsional yang harus dipenuhi adalah *usability*, dan *compability*. Setelah kebutuhan fungsional telah ditentukan maka akan digambarkan dengan *usecase diagram*, *use case scenario*. Pada tahapan analisis kebutuhan terdapat dua kali iterasi. Iterasi yang pertama adalah penggabungan fitur riwayat KB dan kalender KB, sehingga fitur riwayat KB dihapus dari sistem. Dan penambahan fitur

rekomendasi KB. Pada iterasi ke dua terdapat penambahan tombol mencari rumah sakit pada halaman isi pemberitahuan KB.

Untuk memberikan sebuah rekomendasi program KB, pada penelitian ini menggunakan metode KNN (*K-Nearest Neighbor*). Metode KNN digunakan dengan membandingkan hasil antara data latih dengan data uji kontrasepsi kemudian dihitung dengan menggunakan rumus KNN. Berdasarkan dari penelitian yang telah dilakukan sebelumnya oleh Nugroho maka didapatkan hasil akurasi sebesar 95%. Prinsip kerja KNN adalah mencari jarak terdekat antara data yang akan dievaluasi dengan K tetangga terdekatnya dalam data latih, terdapat Sembilan kategori yang digunakan untuk menentukan jarak yaitu umur istri, pendidikan istri, pendidikan suami, jumlah anak, agama istri, pekerjaan istri, pekerjaan suami, standart kehidupan, dan informasi kontrasepsi. Hasil dari perhitungan KNN ini diidentifikasi menjadi tiga kelas, yaitu tidak menggunakan, jangka panjang, dan jangka pendek.

3.3 Perancangan Sistem

Setelah mendapatkan hasil dari analisis kebutuhan yang telah dilakukan dengan proses UCD pada tahapan *understand and specify the context of use* dan *specify the user and organizational requirements*, maka akan masuk ke tahapan UCD *produce design solution*. Pada tahapan ini hasil dari analisis kebutuhan yang sudah didapatkan akan dimasukkan ke dalam perancangan sistem. Pada tahapan perancangan sistem ini dibagi menjadi empat bagian, berikut adalah penjelasan tahapan dari perancangan sistem.

1. Perancangan Data

Pada tahap ini dilakukan perancangan *database* yang digunakan untuk menyimpan seluruh data informasi yang dibutuhkan oleh sistem. Perancangan data ini membutuhkan pembuatan kelas diagram sebagai acuan untuk membuat struktur *database* sistem. Setelah itu akan dispesifikasikan dalam Tabel untuk menyimpan data yang digunakan untuk menyimpan informasi pada aplikasi program KB.

2. Perancangan Arsitektur

Pada tahap ini dilakukan perancangan bagaimana kerangka serta alur sistem akan dikembangkan. Perancangan arsitektur ini terdiri dari *class diagram* dan *sequence diagram*.

3. Perancangan Antarmuka

Perancangan antarmuka dibuat sesuai dengan hasil analisis kebutuhan serta solusi dari masalah. Pada perancangan antarmuka penulis menggunakan simulasi *prototype* untuk memperjelas spesifikasi kebutuhan yang diinginkan oleh pengguna. Jenis *prototype* yang penulis gunakan adalah *paper prototyping*.

4. Perancangan Algoritme

Pada tahap ini dilakukan perancangan algoritme dari sistem yang akan dikembangkan. Perancangan algoritme didasarkan pada kebutuhan fungsional yang telah didefinisikan. Algoritme ini direpresentasikan dalam bentuk *pseudocode* yang kemudian akan di implementasikan ke dalam program.

3.4 Evaluasi Pengguna

Setelah semua tahapan UCD dari *specify the context of use, specify the user and organizational requirements, produce design solution* maka masuk ke tahapan *evaluate design against requirements*. Tahapan ini digunakan untuk melakukan pengukuran apakah rancangan yang peneliti buat sudah dibangun dengan sesuai kebutuhan yang didapatkan dari analisis yang telah dilakukan sebelumnya. Ketika sudah sesuai maka masuk ke tahap implementasi, namun jika belum sesuai dengan kebutuhan yang didapatkan, maka akan kembali ke tahapan *specify the context of use*, dan seterusnya sehingga pengguna sudah puas terhadap aplikasi yang dibuat. Proses evaluasi pengguna ini dilakukan dengan wawancara langsung kepada pengguna KB.

3.5 Implementasi Sistem

Implementasi sistem adalah tahapan dimana dilakukannya eksekusi terhadap data-data penelitian yang telah sesuai dengan kebutuhan yang diinginkan oleh pengguna. Setelah melakukan iterasi dan sudah mencapai hasil akhir maka hasil dari akhir iterasi tersebut di implementasikan kedalam Bahasa pemrograman java berbasis Android untuk membuat aplikasi, dan menggunakan *firebase* serta *sqlite database* untuk menyimpan data. Implementasi sistem diawali dengan mengetahui spesifikasi dari perangkat keras, spesifikasi perangkat lunak, spesifikasi perangkat bergerak. Setelah mengetahui spesifikasi yang digunakan untuk membangun aplikasi KBku maka selanjutnya adalah menentukan batasan implementasi. Kemudian melakukan implementasi basis data, implementasi *class*, implementasi kode program, dan yang terakhir adalah implementasi antarmuka.

3.6 Pengujian Sistem

Pada tahapan pengujian sistem dilakukan untuk menguji apakah sistem yang telah dibuat telah sesuai dengan apa yang pengguna harapkan, dan telah memenuhi kebutuhan-kebutuhan yang telah di analisis. Pada proses pengujian sistem ini maka dapat diketahui apa yang menjadi kelemahan serta kesalahan yang dimiliki oleh sistem agar dapat mendapatkan perbaikan segera oleh peneliti, perbaikan yang peneliti lakukan agar menghasilkan sistem yang sesuai dengan keinginan dan kebutuhan pengguna. Pengujian dalam penelitian ini terbagi menjadi dua pengujian, yaitu pengujian fungsional dan pengujian non fungsional. Pada pengujian fungsional terdiri dari pengujian unit dan pengujian validasi. Sedangkan pengujian non fungsional terdiri dari pengujian *usability* dan pengujian

compatibility. Pengujian *usability* dilakukan untuk menguji apakah sistem yang telah dikembangkan mudah untuk dipelajari dan mudah untuk digunakan. Dan yang terakhir pengujian *compatibility* yang digunakan untuk menguji apakah aplikasi yang sudah dikembangkan dapat berjalan dengan baik di *smartphone* yang berbeda.

3.7 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dan saran adalah proses terakhir dari pembuatan sistem ketika sistem yang dibuat telah jadi dengan sempurna dan sesuai dengan apa yang dibutuhkan. Pengambilan kesimpulan diambil dari hasil pengujian serta analisis sistem yang telah dilakukan pada tahapan sebelumnya. Kemudian penulisan saran digunakan untuk memperbaiki kesalahan yang ada, harapannya dapat menyempurnakan penulisan dan dapat memberikan masukan bagi peneliti selanjutnya.



BAB 4 REKAYASA KEBUTUHAN

Pada bab empat, rekayasa kebutuhan menjelaskan tentang analisis yang dilakukan dalam membangun aplikasi KBku. Pertama peneliti melakukan pengumpulan data yang digunakan untuk mencari tahu kebutuhan dari pengguna aplikasi yaitu wanita usia subur yang sedang atau akan menggunakan KB. Dari hasil pengumpulan data tersebut maka didapatkan sebuah kebutuhan fungsional dan non fungsional, kemudian kebutuhan tersebut digunakan untuk membuat perancangan perangkat lunak yang digambarkan melalui *use case diagram* dan *use case scenario*.

4.1 Analisis Kebutuhan

Analisis kebutuhan digunakan untuk mencari kebutuhan pengguna aplikasi pada awal proses untuk membuat sebuah aplikasi berdasarkan metode UCD. Tahapan pada UCD yang digunakan untuk analisa kebutuhan adalah *specify the context of use* dan *specify user and organizational requirements* yang berarti pada tahap ini dilakukan proses tentang gambaran umum sistem, identifikasi aktor, serta penjabaran kebutuhan fungsional maupun kebutuhan non-fungsional. Setelah tahapan analisis kebutuhan telah selesai dilakukan maka hasil dari analisis akan di gambarkan dalam bentuk *use case diagram* dan *use case scenario*. Kemudian hasil dari analisis tersebut akan masuk ke tahapan perancangan. Di tahapan perancangan ini akan dilakukan evaluasi apakah analisa kebutuhan yang sudah dibuat dan masuk pada tahapan perancangan sesuai dengan apa yang pengguna aplikasi inginkan, jika masih belum sesuai maka peneliti akan mendatangi kembali calon pengguna aplikasi untuk melihat apakah kebutuhan yang sudah di analisis telah sesuai dengan apa yang calon pengguna aplikasi inginkan, jika masih belum sesuai, maka akan ada analisis kebutuhan iterasi kedua dan seterusnya sampai sesuai dengan apa yang calon pengguna KB butuhkan.

4.1.1 Gambaran Umum Aplikasi

Aplikasi KBku merupakan aplikasi berbasis android yang digunakan untuk membantu pengguna KB untuk lebih memahami program keluarga berencana dan menekan tingkat kegagalan dalam menggunakan KB. Harapannya pengguna KB tidak lagi gagal dalam menggunakan KB dan lebih mengerti tentang perawatan, penggunaan, serta informasi terkait program KB yang dilaksanakan oleh pemerintah.

4.1.1.1 Deskripsi Aplikasi



Gambar 4.1 Gambar Deskripsi Aplikasi

Gambar 4.1 merupakan aplikasi KBku ini adalah sebuah aplikasi yang digunakan untuk membantu para wanita usia subur untuk mempermudah melakukan program KB dan juga memberikan informasi tentang program KB yang cocok dengan segala kebutuhan pengguna KB. Untuk menggunakan aplikasi KBku ini, pengguna harus terhubung dengan jaringan *internet* agar dapat melakukan operasi pada fitur-fitur yang ada pada aplikasi KBku. Aplikasi perangkat bergerak program KB KBku ini memiliki dua pengguna yaitu pengguna aplikasi yang belum terdaftar ke dalam sistem dan belum bisa menggunakan fitur yang terdapat pada aplikasi KBKU, yaitu pengguna KB. Sebelum pengguna KB masuk ke dalam aplikasi dan mulai menggunakan fitur yang telah disediakan, maka pengguna diharuskan untuk melakukan pendaftaran akun terlebih dahulu, setelah melakukan pendaftaran akun, maka pengguna akan mengisi nama pengguna dan kata sandi yang telah di daftarkan sebelumnya untuk bisa masuk ke dalam aplikasi KBku ini. Ketika pengguna KB sudah selesai mendaftarkan akunnya, maka pengguna KB berubah menjadi Akseptor. Akseptor disini adalah pengguna aplikasi yang sudah memiliki akun serta dapat melakukan operasi pada fitur-fitur yang terdapat pada aplikasi KBku. Akseptor dapat melihat informasi seputar program KB. Akseptor dapat menyimpan riwayat penggunaan KB sebelumnya. Akseptor dapat membuat waktu kembali untuk melakukan KB sehingga ketika saat melakukan KB selanjutnya, akseptor mendapatkan pemberitahuan untuk melakukan program KB. Dan akseptor juga dapat melakukan pencarian rumah sakit terdekat yang terdapat pada aplikasi KBKU.

4.1.1.2 Identifikasi Aktor

Pada tahap identifikasi aktor dilakukan identifikasi terhadap pengguna yang nantinya terlibat di dalam sistem aplikasi yang dibuat. Di dalam aplikasi program KB KBku, terdapat dua aktor, yaitu pengguna KB dan akseptor. Penjelasan lengkap mengenai deskripsi pengguna KB dan akseptor dapat dilihat di Tabel 4.1.

Tabel 4.1 Identifikasi Aktor

Nama Aktor	Deskripsi
Pengguna KB	Pengguna KB merupakan pengguna yang masih belum mendaftarkan akun kedalam aplikasi.
Akseptor	Merupakan pengguna aplikasi yang telah daftar kedalam sistem. Akseptor dapat melihat informasi terkait KB, serta dapat melihat letak rumah sakit terdekat. Akseptor akan mendapatkan pemberitahuan KB terkait tanggal kontrol balik untuk melakukan program KB selanjutnya. Akseptor juga dapat menyimpan seluruh riwayat KB yang telah akseptor gunakan sebelumnya.

4.1.1.3 Analisis Kebutuhan Fungsional

Pada daftar kebutuhan fungsional pada penelitian ini dikhususkan untuk pengguna aplikasi yaitu wanita usia subur yang sedang atau akan memakai KB yang belum mendaftarkan diri pada aplikasi, dan akseptor yaitu pengguna aplikasi yang telah mendaftarkan diri pada aplikasi KBku dan bisa menjalankan fitur-fitur pada aplikasi KBku. Pada subbab kebutuhan fungsional disini mencakup kode fungsi, nama fungsi, serta deskripsi dari setiap kebutuhan fungsional. Berikut merupakan penjelasan setiap kebutuhan fungsional untuk pengguna KB dan akseptor.

Pada lampiran berikut merupakan hasil wawancara yang telah dilakukan terhadap pengguna KB. Hasil dari wawancara yang telah dilakukan dapat dilihat pada Tabel 4.2.

Tabel 4.2 Identifikasi Kebutuhan Fungsional Pengguna KB

No	Nama Kebutuhan	Jumlah Orang	Presentase (%)
1	Pengguna KB membutuhkan informasi KB	Setuju : 6 Tidak Setuju : 0	100% 0%
2	Pengguna KB membutuhkan pemberitahuan untuk kontrol balik	Setuju : 6 Tidak Setuju : 0	100% 0%
3	Pengguna KB membutuhkan penyimpanan untuk segala informasi terkait data KB yang pernah atau akan digunakan	Setuju : 6 Tidak Setuju : 0	100% 0%
4	Pengguna KB membutuhkan untuk mencari rumah sakit terdekat	Setuju : 6 Tidak Setuju : 0	100% 0%

A. Kebutuhan Fungsional Pengguna KB

Kebutuhan fungsional bagi pengguna KB merupakan kebutuhan yang dibutuhkan oleh pengguna terhadap aplikasi program KB yang nantinya digunakan untuk membantu pengguna dalam melaksanakan program KB serta dapat melakukan operasi pada fitur-fitur yang terdapat pada aplikasi KBku. Berikut adalah kebutuhan fungsional pengguna, dapat dilihat di Tabel 4.3

Tabel 4.3 Identifikasi Kebutuhan Fungsional Pengguna KB

No.	Kode Fungsi	Nama Fungsi	Deskripsi
1	SRS – F – 01	Daftar Akun	Pengguna KB dapat membuat akun pada aplikasi program KB

2	SRS – F – 02	Masuk ke sistem	Pengguna KB dapat melakukan masuk kedalam aplikasi program KB
---	--------------	-----------------	---

B. Kebutuhan Fungsional Akseptor

Kebutuhan fungsional bagi akseptor merupakan kebutuhan yang dibutuhkan oleh pengguna terhadap aplikasi program KB yang nantinya digunakan untuk membantu pengguna dalam melaksanakan program KB ketika pengguna KB telah memiliki akun pada aplikasi KBku. Berikut adalah kebutuhan fungsional akseptor, dapat dilihat di Tabel 4.4.

Tabel 4.4 Identifikasi Kebutuhan Fungsional Akseptor

No.	Kode Fungsi	Nama Fungsi	Deskripsi
1	SRS – F – 03	Melihat Informasi KB	Akseptor dapat melihat informasi KB
2	SRS – F – 04	Melihat Riwayat KB	Akseptor dapat melihat riwayat KB
3	SRS – F – 05	Membuat Riwayat KB	Akseptor dapat membuat riwayat KB
4	SRS – F – 06	Menghapus Riwayat KB	Akseptor dapat menghapus riwayat KB yang telah dibuat
5	SRS – F – 07	Melihat Riwayat Kalender KB	Akseptor dapat melihat riwayat kalender KB
6	SRS – F – 08	Membuat Kalender KB	Akseptor dapat membuat kalender KB
7	SRS – F – 09	Menerima Pemberitahuan KB	Akseptor dapat menerima pemberitahuan KB
8	SRS – F – 10	Melihat Pemberitahuan KB	Akseptor dapat melihat pemberitahuan KB
9	SRS – F – 11	Mencari Rumah Sakit	Akseptor dapat mencari rumah sakit terdekat
10	SRS – F – 12	Melihat Profil	Akseptor dapat melihat profil
11	SRS – F – 13	Mengubah Profil	Akseptor dapat mengubah profil
12	SRS – F – 14	Keluar dari Sistem	Akseptor dapat keluar dari sistem

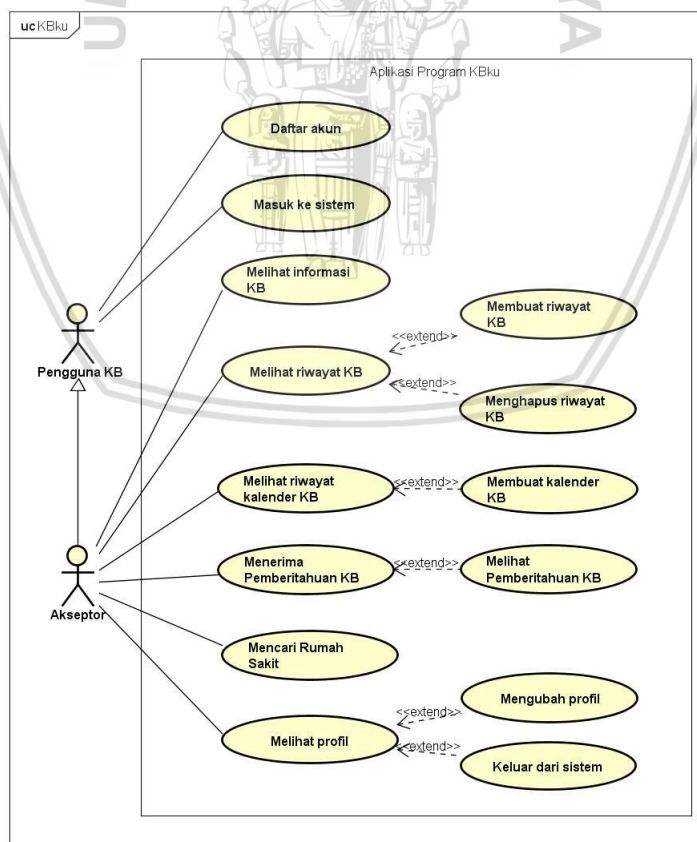
4.1.1.4 Analisis Kebutuhan Non-Fungsional

Kebutuhan non fungsional merupakan kebutuhan yang digunakan untuk mengembangkan aplikasi yang mencakup kualitas dan batasan sistem. Berikut adalah penjabaran kebutuhan non fungsional aplikasi KBKU yang terdapat pada Tabel 4.5.

Tabel 4.5 Identifikasi Kebutuhan Fungsional Akseptor

No.	Parameter	Deskripsi Kebutuhan
1	<i>Usability</i>	Aplikasi KBku harus mudah digunakan oleh pengguna, pengguna KBku harus mengetahui setiap fungsi fitur dengan baik. Serta harus memenuhi parameter <i>learnability</i> , <i>efficiency</i> , <i>memorability</i> , <i>errors</i> , dan <i>satisfaction</i> .
2	<i>Compatibility</i>	Aplikasi KBku dapat memenuhi kebutuhan <i>compatibility</i> yang dapat diuji dengan menggunakan beberapa <i>smartphone</i> yang sesuai dengan spesifikasi aplikasi yang sedang dikembangkan.

4.1.1.5 Use Case Diagram



Gambar 4.2 Gambar Use Case Diagram

Pada Gambar 4.2 menjelaskan tentang *use case diagram* pengguna aplikasi program KB dan kebutuhan yang di perlukan oleh pengguna. Pada sistem aplikasi KBku, dimana dijelaskan bahwa terdapat dua pengguna aplikasi yaitu pengguna KB dan akseptor. Pengguna KB merupakan pengguna aplikasi yang belum mendaftarkan diri pada aplikasi KB, sehingga pengguna KB belum memiliki hak akses terhadap fitur-fitur yang ada pada aplikasi KBku. Pengguna KB diharuskan untuk mendaftarkan diri dengan cara daftar akun dan kemudian masuk kedalam sistem aplikasi KBku. Pengguna aplikasi yang kedua adalah akseptor. Akseptor merupakan turunan dari pengguna KB, dimana ketika pengguna KB telah sukses mendaftarkan akun dan dapat masuk ke dalam sistem. Pengguna KB akan berubah menjadi akseptor dan dapat melakukan operasi pada fitur-fitur yang telah disediakan oleh sistem yaitu melihat informasi KB, melihat riwayat KB, pada saat melihat riwayat KB akseptor dapat melakukan pembuatan riwayat KB baru dan menghapus riwayat KB yang sudah dibuat sebelumnya. Melihat riwayat kalender KB, ketika akseptor melihat riwayat kalender KB maka akseptor juga dapat memilih pilihan membuat kalender KB. Menerima pemberitahuan KB, ketika akseptor mendapatkan pemberitahuan untuk melakukan program KB, maka akseptor dapat melihat isi dari pemberitahuan KB yang telah didapatkan. Akseptor dapat mencari rumah sakit terdekat. Akseptor dapat melihat profil yang telah dibuat sebelumnya dan dapat mengubah profil tersebut. Kemudian akseptor dapat keluar dari sistem.

4.1.1.6 Use Case Scenario

Dalam *use case scenario* berisi tentang penjelasan kebutuhan pengguna yang lebih detail. Pada *use case scenario* diberikan penjelasan yang berisi nama *use case*, pengguna (*actor*), kondisi awal (*Pre-Condition*), aksi yang dilakukan (*Main Flow*), kondisi alternatif (*alternatife flow*), dan kondisi yang harus di penuhi oleh sistem (*Post-Condition*). Berikut ini adalah penjabaran dari *use case scenario* yang dapat dilihat pada penjelasan dibawah ini.

A. Use Case Scenario Pengguna KB

1. Use Case Scenario Pengguna KB Daftar Akun

Berikut ini adalah *use case scenario* pengguna KB untuk melakukan proses daftar akun sebelum bisa masuk kedalam sistem. Penjelasan lengkap tentang pengguna KB melakukan proses daftar akun dijelaskan pada Tabel 4.6.

Tabel 4.6 Use Case Scenario Pengguna KB Daftar Akun

<i>Objective</i>	Pengguna KB melakukan pendaftaran akun baru
<i>Actor</i>	Pengguna KB
<i>Pre-condition</i>	Pengguna KB membuka aplikasi KB dan memilih menu daftar
<i>Main flow</i>	1. Pengguna KB memilih menu daftar pada halaman awal aplikasi

	<ol style="list-style-type: none"> 2. Sistem akan menampilkan form yang harus diisi oleh Pengguna KB 3. Pengguna KB mengisi data diri sesuai dengan kolom-kolom yang ada pada halaman daftar 4. Pengguna KB memilih tombol “daftar” untuk menyimpan data 5. Data tersimpan pada <i>database</i> aplikasi KBku 6. Sistem menampilkan halaman masuk
<i>Alternative flows</i>	<ol style="list-style-type: none"> 7. Jika proses pendaftaran akun baru gagal maka akan muncul pemberitahuan KB “maaf tidak bisa menyimpan data coba kembali” 8. Sistem menampilkan halaman daftar
<i>Post-condition</i>	Pengguna KB berhasil daftar akun baru

Pada Tabel 4.6 dijelaskan bahwa sebelum pengguna KB menggunakan aplikasi KB, maka pengguna diharuskan untuk mendaftarkan akun terlebih dahulu dengan mengisi data yang sesuai dengan form yang terdapat pada halaman daftar akun. Kemudian pengguna KB memilih tombol daftar agar data yang sudah diisikan tersimpan didalam *database* aplikasi KBku, kemudian sistem akan menampilkan halaman masuk. Namun jika gagal dalam melakukan penyimpanan, maka aplikasi menampilkan pemberitahuan dan akan menampilkan kembali halaman daftar.

2. Use Case Scenario Pengguna KB Masuk ke sistem

Berikut ini adalah *use case scenario* pengguna KB untuk melakukan proses masuk ke dalam sistem aplikasi. Penjelasan lengkap tentang pengguna KB melakukan proses masuk ke dalam sistem dijelaskan pada Tabel 4.7.

Tabel 4.7 Use Case Scenario Pengguna KB Masuk ke Sistem

<i>Objective</i>	Pengguna KB dapat masuk ke dalam sistem
<i>Actor</i>	Pengguna KB
<i>Pre-condition</i>	Pengguna KB sudah melakukan daftar akun
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Pengguna KB memasukkan nama pengguna KB yang telah di daftarkan saat mendaftarkan akun 2. Pengguna KB memasukkan kata sandi akun yang sudah di daftarkan saat mendaftarkan akun 3. Pengguna KB memilih tombol “masuk” untuk masuk ke dalam sistem aplikasi 4. Pengguna KB masuk ke dalam sistem
<i>Alternative flows</i>	<ol style="list-style-type: none"> 5. Jika Pengguna KB salah memasukkan nama Pengguna KB atau kata sandi sistem akan menampilkan “nama Pengguna KB atau kata sandi salah”

	6. Jika proses pendaftaran akun baru gagal maka akan muncul pemberitahuan KB “maaf tidak bisa menyimpan data coba kembali” 7. Sistem akan menampilkan form halaman masuk
<i>Post-condition</i>	Pengguna KB dapat masuk ke dalam sistem

Pada Tabel 4.7 dijelaskan bahwa setelah pengguna KB melakukan pendaftaran akun, maka pengguna KB harus mengisi data yang telah di daftarkan sebelumnya pada aplikasi. Pengguna KB harus memasukkan nama pengguna, serta kata sandi yang telah terdaftar, kemudian memilih tombol masuk. Jika pengguna KB salah memasukkan kata sandi, nama pengguna, atau tidak dapat masuk ke dalam sistem maka muncul pemberitahuan dan sistem akan menampilkan kembali halaman masuk, sehingga pengguna KB harus memasukkan ulang nama pengguna dan kata sandi yang sesuai dengan data yang sudah terdaftar sebelumnya.

A. Use Case Scenario Akseptor

1. Use Case Scenario Akseptor Melihat Informasi KB

Berikut ini adalah *use case scenario* akseptor melihat informasi seputar KB. Penjelasan lengkap tentang akseptor melihat informasi KB dijelaskan pada Tabel 4.8.

Tabel 4.8 Use Case Scenario Akseptor Melihat Informasi KB

<i>Objective</i>	Akseptor Melihat Informasi KB
<i>Actor</i>	Akseptor
<i>Pre-condition</i>	Akseptor sudah memiliki akun pada aplikasi KBku, dan telah berhasil masuk kedalam aplikasi KBku
<i>Main flow</i>	1. Akseptor memilih menu informasi KB 2. Sistem akan menampilkan halaman tentang informasi KB 3. Akseptor dapat melihat informasi KB
<i>Alternative flows</i>	4. Jika tidak ada koneksi <i>internet</i> maka sistem akan menampilkan pesan “periksa jaringan <i>internet</i> anda”
<i>Post-condition</i>	Akseptor dapat melihat informasi KB

Pada Tabel 4.8 dijelaskan bahwa akseptor dapat melihat informasi KB. Untuk melihat informasi KB, akseptor sudah harus memiliki akun, dan berhasil masuk kedalam aplikasi KBku. Untuk melihat informasi KB, akseptor memilih menu informasi KB yang terdapat pada halaman beranda, kemudian sistem akan

menampilkan berbagai macam informasi-informasi seputar program KB. Jika gagal dalam menampilkan informasi, sistem akan muncul pemberitahuan.

2. *Use Case Scenario* Akseptor Melihat Riwayat KB

Berikut ini adalah *use case scenario* akseptor melihat riwayat pemakaian KB. Penjelasan lengkap tentang akseptor melihat riwayat pemakaian KB dijelaskan pada Tabel 4.9.

Tabel 4.9 *Use Case Scenario* Akseptor Melihat Riwayat KB

<i>Objective</i>	Akseptor Melihat Riwayat Pemakaian KB
<i>Actor</i>	Akseptor
<i>Pre-condition</i>	Akseptor sudah memiliki akun pada aplikasi KBku, dan telah berhasil masuk kedalam aplikasi KBku
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Akseptor masuk ke halaman beranda 2. Akseptor memilih menu "Riwayat KB" 3. Akseptor memilih riwayat KB yang sudah dibuat sebelumnya 4. Akseptor dapat melihat riwayat KB
<i>Alternative flows</i>	5. Jika sistem gagal menampilkan riwayat KB maka muncul pemberitahuan "periksa jaringan <i>internet</i> anda"
<i>Post-condition</i>	Akseptor dapat melihat riwayat KB

Pada Tabel 4.9 dijelaskan bahwa akseptor melihat riwayat KB. Untuk melihat riwayat KB, akseptor sudah harus memiliki akun dan berhasil masuk kedalam aplikasi KBku. Kemudian akseptor memilih menu riwayat KB, dan melihat daftar riwayat KB yang telah dibuat sebelumnya, selanjutnya akseptor memilih riwayat KB mana yang ingin dilihat. Jika gagal melihat riwayat KB, maka akan muncul pemberitahuan.

3. *Use Case Scenario* Akseptor Membuat Riwayat KB

Berikut ini adalah *use case scenario* akseptor membuat riwayat pemakaian KB. Penjelasan lengkap tentang akseptor membuat riwayat pemakaian KB dijelaskan pada Tabel 4.10.

Tabel 4.10 *Use Case Scenario* Akseptor Membuat Riwayat KB

<i>Objective</i>	Akseptor Membuat Riwayat Pemakaian KB
<i>Actor</i>	Akseptor
<i>Pre-condition</i>	Akseptor sudah memiliki akun pada aplikasi KBku, dan telah berhasil masuk kedalam aplikasi KBku
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Akseptor masuk ke halaman beranda 2. Akseptor memilih menu "Riwayat KB"

	<ol style="list-style-type: none"> 3. Akseptor memilih menu “Riwayat KB Baru ?” 4. Akseptor memasukkan informasi yang sudah tersedia pada form riwayat KB 5. Akseptor memilih tombol “simpan” 6. Sistem akan melakukan penyimpanan terkait data riwayat akseptor 7. Akseptor dapat melakukan pembuatan riwayat KB dan kembali ke halaman riwayat KB
<i>Alternative flows</i>	<ol style="list-style-type: none"> 8. Jika penyimpanan data akseptor tidak tersimpan maka akseptor akan kembali ke halaman riwayat KB dan muncul pemberitahuan “gagal menyimpan data”
<i>Post-condition</i>	Akseptor dapat membuat riwayat KB

Pada Tabel 4.10 dijelaskan akseptor membuat riwayat KB. Untuk membuat riwayat KB, akseptor sudah harus masuk ke sistem dan berhasil masuk ke dalam sistem. Akseptor akan masuk ke halaman beranda, kemudian akseptor memilih menu riwayat KB dan membuat riwayat KB baru, kemudian memasukkan informasi sesuai dengan yang sudah disediakan oleh form aplikasi. Ketika sudah selesai memasukkan data, maka akseptor memilih tombol “simpan”. Jika gagal melakukan penyimpanan, maka akseptor akan menerima pemberitahuan dan kembali ke halaman riwayat KB.

4. *Use Case Scenario* Akseptor Menghapus Riwayat KB

Berikut ini adalah *use case scenario* akseptor menghapus riwayat pemakaian KB. Penjelasan lengkap tentang akseptor hapus riwayat pemakaian KB dijelaskan pada Tabel 4.11.

Tabel 4.11 *Use Case Scenario* Akseptor menghapus Riwayat KB

<i>Objective</i>	Akseptor Menghapus Riwayat Pemakaian KB
<i>Actor</i>	Akseptor
<i>Pre-condition</i>	Akseptor sudah memiliki akun pada aplikasi KBku, dan telah berhasil masuk ke dalam aplikasi KBku
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Akseptor masuk ke halaman beranda 2. Akseptor memilih menu “Riwayat KB” 3. Akseptor dapat melihat daftar riwayat KB yang sudah dibuat oleh akseptor 4. Akseptor memilih salah satu dari daftar riwayat KB yang sudah akseptor buat 5. Akseptor masuk ke halaman detail riwayat KB, dan memilih Gambar tempat sampah untuk melakukan hapus data riwayat KB

	6. Sistem akan melakukan perubahan data pada <i>database</i> 7. Perubahan telah selesai dibuat oleh akseptor dan kembali ke halaman riwayat KB
<i>Alternative flows</i>	8. Jika sistem gagal menghapus riwayat KB maka muncul pemberitahuan “periksa jaringan <i>internet</i> anda”
<i>Post-condition</i>	Akseptor dapat hapus riwayat KB

Pada Tabel 4.11 dijelaskan bahwa akseptor dapat menghapus riwayat KB. Untuk dapat menghapus riwayat KB, akseptor harus sudah memiliki akun dan berhasil masuk kedalam aplikasi KBku. Akseptor memilih menu riwayat KB, kemudian memilih salah satu daftar riwayat KB yang telah dibuat, kemudian akseptor memilih tombol hapus yang bergambar tempat sampah untuk melakukan hapus riwayat KB. Jika gagal melakukan hapus riwayat KB maka sistem akan menampilkan pemberitahuan.

5. *Use Case Scenario* Akseptor Melihat Riwayat Kalender KB

Berikut ini adalah *use case scenario* akseptor melihat riwayat kalender KB. Penjelasan lengkap tentang akseptor melihat riwayat kalender KB dijelaskan pada Tabel 4.12.

Tabel 4.12 *Use Case Scenario* Akseptor Melihat Riwayat Kalender KB

<i>Objective</i>	Akseptor Melihat Riwayat Kalender KB
<i>Actor</i>	Akseptor
<i>Pre-condition</i>	Akseptor sudah memiliki akun pada aplikasi KBku, dan telah berhasil masuk kedalam aplikasi KBku
<i>Main flow</i>	1. Akseptor masuk ke halaman beranda 2. Akseptor memilih menu “Kalender KB” pada halaman beranda
<i>Alternative flows</i>	-
<i>Post-condition</i>	Akseptor dapat melihat riwayat KB yang telah dibuat

Pada Tabel 4.12 dijelaskan bahwa akseptor dapat dapat melihat riwayat kalender KB. Untuk dapat melihat riwayat kalender KB, akseptor harus sudah memiliki akun dan berhasil masuk kedalam aplikasi KBku. Akseptor memilih menu kalender KB. Sistem akan menampilkan daftar dari kalender KB yang sudah dilakukan oleh akseptor sebelumnya.

6. *Use Case Scenario* Akseptor Membuat Kalender KB

Berikut ini adalah *use case scenario* akseptor membuat kalender KB. Penjelasan lengkap tentang akseptor membuat kalender KB dijelaskan pada Tabel 4.13.

Tabel 4.13 *Use Case Scenario* Akseptor Membuat Kalender KB

<i>Objective</i>	Akseptor Membuat Kalender KB
<i>Actor</i>	Akseptor
<i>Pre-condition</i>	Akseptor sudah memiliki akun pada aplikasi KBku, dan telah berhasil masuk kedalam aplikasi KBku
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Akseptor memilih menu kalender KB pada halaman beranda 2. Akseptor memilih metode KB 3. Akseptor memasukkan tanggal KB terakhir 4. Akseptor memasukkan berat badan 5. Akseptor memilih tombol buat pemberitahuan
<i>Alternative flows</i>	6. Jika ada data yang belum diisi maka sistem akan menampilkan pemberitahuan “isikan data dengan lengkap”
<i>Post-condition</i>	Akseptor dapat membuat kalender KB

Pada Tabel 4.13 dijelaskan bahwa akseptor dapat membuat kalender KB. Untuk dapat membuat kalender KB, akseptor harus sudah memiliki akun dan berhasil masuk kedalam aplikasi KBku. Akseptor memilih menu kalender KB. Akseptor memilih metode KB yang digunakan, memasukkan jangka waktu pemberitahuan, serta berat badan terakhir dan memilih tombol buat pemberitahuan. Namun jika proses penyimpanan gagal maka akan muncul pemberitahuan.

7. *Use Case Scenario* Akseptor Menerima Pemberitahuan KB

Berikut ini adalah *use case scenario* akseptor menerima pemberitahuan KB. Penjelasan lengkap tentang akseptor menerima pemberitahuan KB dijelaskan pada Tabel 4.14.

Tabel 4.14 *Use Case Scenario* Akseptor Menerima Pemberitahuan KB

<i>Objective</i>	Akseptor Menerima Pemberitahuan KB
<i>Actor</i>	Akseptor
<i>Pre-condition</i>	Akseptor sudah memiliki akun pada aplikasi KBku, dan telah berhasil masuk kedalam aplikasi KBku

<i>Main flow</i>	<ol style="list-style-type: none"> 1. Akseptor telah memasukkan dan menyimpan data untuk membuat kalender KB pada halaman buat kalender KB 2. Jika sudah pada tanggal untuk melakukan KB maka Akseptor mendapatkan pemberitahuan KB
<i>Alternative flows</i>	-
<i>Post-condition</i>	Akseptor menerima pemberitahuan KB

Pada Tabel 4.14 dijelaskan bahwa akseptor dapat menerima pemberitahuan KB. Untuk dapat melihat pemberitahuan kalender KB, akseptor harus sudah memiliki akun dan berhasil masuk kedalam aplikasi KBku. Jika sudah pada tanggal yang ditetapkan, maka akseptor akan menerima pemberitahuan KB diaplikasinya.

8. *Use Case Scenario* Akseptor Melihat Pemberitahuan KB

Berikut ini adalah *use case scenario* akseptor melihat pemberitahuan KB. Penjelasan lengkap tentang akseptor melihat pemberitahuan KB dijelaskan pada Tabel 4.15.

Tabel 4.15 *Use Case Scenario* Akseptor Melihat Pemberitahuan KB

<i>Objective</i>	Akseptor Melihat Pemberitahuan KB
<i>Actor</i>	Akseptor
<i>Pre-condition</i>	Akseptor sudah memiliki akun pada aplikasi KBku, dan telah berhasil masuk kedalam aplikasi KBku
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Akseptor telah memasukkan dan menyimpan data untuk membuat kalender KB pada halaman buat kalender KB 2. Jika sudah pada tanggal untuk melakukan KB maka Akseptor mendapatkan pemberitahuan KB 3. Akseptor dapat melihat isi pemberitahuan KB
<i>Alternative flows</i>	-
<i>Post-condition</i>	Akseptor melihat Kalender KB

Pada Tabel 4.15 dijelaskan bahwa akseptor dapat melihat kalender KB. Untuk dapat melihat pemberitahuan kalender KB, akseptor harus sudah memiliki akun dan berhasil masuk kedalam aplikasi KBku. Jika sudah pada tanggal yang ditetapkan, maka akseptor akan menerima pemberitahuan KB di aplikasinya. Kemudian akseptor dapat memilih pemberitahuan tersebut dan sistem akan menampilkan isi dari pemberitahuan KB.

9. Use Case Scenario Akseptor Mencari Rumah sakit

Berikut ini adalah *use case scenario* akseptor untuk melakukan proses pencarian rumah sakit terdekat. Penjelasan lengkap tentang akseptor melakukan proses pencarian rumah sakit terdekat akan dijelaskan pada Tabel 4.16.

Tabel 4.16 Use Case Scenario Akseptor Mencari Rumah sakit

<i>Objective</i>	Akseptor Mencari Rumah sakit Tedekat
<i>Actor</i>	Akseptor Aplikasi
<i>Pre-condition</i>	Akseptor sudah masuk ke dalam sistem, kemudian akseptor memilih menu rumah sakit
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Akseptor memilih menu Rumah Sakit pada halaman beranda 2. Akseptor memilih menu rumah sakit pada halaman beranda 3. Akseptor memasukkan lokasi keberadaannya saat itu 4. Akseptor memilih tombol cari 5. Sistem akan melakukan pencarian terhadap lokasi yang akan dituju 6. Sistem akan menampilkan hasil dari pencarian
<i>Alternative flows</i>	<ol style="list-style-type: none"> 7. Jika tidak ada koneksi <i>internet</i> maka sistem akan menampilkan “periksa jaringan <i>internet</i> anda”
<i>Post-condition</i>	Akseptor dapat melihat rumah sakit terdekat

Pada Tabel 4.16 dijelaskan bahwa akseptor dapat mencari rumah sakit. Untuk dapat mencari rumah sakit, akseptor harus sudah memiliki akun dan berhasil masuk kedalam aplikasi KBku. Akseptor memilih menu rumah sakit pada halaman beranda, kemudian akseptor memasukkan lokasi keberadaannya sekarang, kemudian akseptor memilih tombol cari, dan sistem akan menampilkan rumah sakit terdekat yang ada di sekitarnya. Namun jika tidak dapat menampilkan rumah sakit, maka akan muncul pemberitahuan.

10. Use Case Scenario Akseptor Melihat Profil

Berikut ini adalah *use case skenario* akseptor melihat profil. Penjelasan lengkap tentang akseptor melihat profil akan dijelaskan pada Tabel 4.17.

Tabel 4.17 Use Case Scenario Akseptor Melihat Profil

<i>Objective</i>	Akseptor Melihat Profil
<i>Actor</i>	Akseptor
<i>Pre-condition</i>	Akseptor sudah memiliki akun pada aplikasi KBku, dan telah berhasil masuk kedalam aplikasi KBku

<i>Main flow</i>	<ol style="list-style-type: none"> 1. Akseptor memilih menu “profil” pada halaman beranda 2. Akseptor melihat profil yang sudah di buat pada saat daftar akun baru
<i>Alternative flows</i>	-
<i>Post-condition</i>	Akseptor dapat melihat profil

Pada Tabel 4.17 dijelaskan bahwa akseptor dapat melihat profil. Untuk dapat melihat profil, akseptor harus sudah memiliki akun dan berhasil masuk kedalam aplikasi KBku. Akseptor memilih menu profil pada halaman beranda kemudian aplikasi akan menampilkan hasil dari profil yang telah dibuat saat mendaftarkan akun.

11. *Use Case Scenario* Akseptor Mengubah Profil

Berikut ini adalah *use case scenario* akseptor mengubah profil. Penjelasan lengkap tentang akseptor mengubah profil akan dijelaskan pada Tabel 4.18.

Tabel 4.18 *Use Case Scenario* Akseptor Mengubah Profil

<i>Objective</i>	Akseptor Mengubah Profil
<i>Actor</i>	Akseptor
<i>Pre-condition</i>	Akseptor sudah memiliki akun pada aplikasi KBku, dan telah berhasil masuk kedalam aplikasi KBku
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Akseptor memilih menu “profil” pada halaman beranda 2. Akseptor memilih tombol ubah profil 3. Sistem menampilkan halaman ubah profil 4. Akseptor merubah data profil sesuai dengan data yang diinginkan 5. Akseptor memilih tombol simpan untuk menyimpan data
<i>Alternative flows</i>	<ol style="list-style-type: none"> 6. Jika proses penyimpanan data tidak dapat dilakukan maka muncul pemberitahuan “Proses penyimpanan gagal” dan kembali ke halaman ubah profil
<i>Post-condition</i>	Akseptor dapat mengubah profil

Pada Tabel 4.18 dijelaskan bahwa akseptor dapat mengubah profil. Untuk dapat mengubah profil, akseptor harus sudah memiliki akun dan berhasil masuk kedalam aplikasi KBku. Akseptor memilih menu profil pada halaman beranda, kemudian akseptor memilih tombol ubah profil. Sistem akan menampilkan halaman ubah profil, kemudian akseptor dapat merubah profil sesuai dengan yang

diinginkan, kemudian akseptor memilih tombol simpan untuk menyimpan data yang telah dirubah. Namun jika penyimpanan yang dilakukan gagal, maka akan muncul pemberitahuan dan kembali ke halaman ubah profil.

12. *Use Case Scenario* Akseptor Keluar dari Sistem

Berikut ini adalah *use case scenario* akseptor untuk keluar dari sistem. Penjelasan lengkap tentang akseptor keluar dari sistem akan dijelaskan pada Tabel 4.19.

Tabel 4.19 *Use Case Scenario* Akseptor Keluar Dari Sistem

<i>Objective</i>	Akseptor keluar dari sistem
<i>Actor</i>	Akseptor Aplikasi
<i>Pre-condition</i>	Akseptor sudah memiliki akun pada aplikasi KBku, dan telah berhasil masuk kedalam aplikasi KBku
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Akseptor memilih menu “profil” pada halaman beranda 2. Akseptor memilih menu “keluar” 3. Sistem akan melakukan proses keluar dari sistem 4. Akseptor keluar dari sistem
<i>Alternative flows</i>	<ol style="list-style-type: none"> 5. Jika tidak ada koneksi <i>internet</i> maka sistem akan menampilkan “periksa jaringan <i>internet</i> anda” dan menampilkan halaman profil
<i>Post-condition</i>	Akseptor dapat keluar dari sistem

Pada Tabel 4.19 dijelaskan bahwa akseptor dapat keluar dari sistem. Untuk dapat keluar dari sistem, akseptor harus sudah memiliki akun dan berhasil masuk kedalam aplikasi KBku. Akseptor memilih menu profil pada halaman beranda, kemudian akseptor memilih menu keluar. Namun jika akseptor gagal keluar dari sistem, maka akan muncul pemberitahuan dan akan menampilkan halaman profil.

4.1.2 Analisis Kebutuhan Iterasi 1

Analisis kebutuhan iterasi pertama merupakan analisis yang dilakukan ketika hasil dari analisis yang dilakukan sebelumnya terdapat perubahan. Analisis kebutuhan pada iterasi pertama dilakukan dengan wawancara secara langsung dengan calon pengguna KB serta bidan, untuk mengetahui kebutuhan apa saja yang benar-benar sesuai dengan kebutuhan calon pengguna KB.

4.1.2.1 Analisis Kebutuhan Fungsional Iterasi 1

Pada analisis kebutuhan pertama, adanya penambahan fitur yang digunakan untuk memberikan rekomendasi untuk calon pengguna KB, berdasarkan wawancara yang dilakukan kepada bidan dan calon pengguna KB, kebutuhan rekomendasi KB merupakan hal yang cukup penting agar pengguna KB dapat mengetahui lebih lanjut tentang program KB yang sesuai dengan kondisinya. Hasil

akhir dari wawancara evaluasi kebutuhan adalah dihapuskannya kebutuhan untuk melihat, membuat, menghapus riwayat KB dikarenakan kebutuhan tersebut bisa digabung dengan kebutuhan kalender KB. Dan juga ditambahkan kebutuhan rekomendasi KB, agar pengguna KB mendapatkan rekomendasi KB yang sesuai dengan kebutuhannya.

A. Kebutuhan Fungsional Akseptor

Kebutuhan fungsional bagi akseptor merupakan kebutuhan yang dibutuhkan oleh pengguna terhadap aplikasi program KB yang nantinya digunakan untuk membantu pengguna dalam melaksanakan program KB ketika pengguna KB telah memiliki akun pada aplikasi KBku. Berikut adalah kebutuhan fungsional akseptor, dapat dilihat di Tabel 4.18.

Tabel 4.20 Identifikasi Kebutuhan Fungsional Pengguna KB

No.	Kode Fungsi	Nama Fungsi	Deskripsi
1	SRS – F – 14	Membuat rekomendasi KB	Akseptor dapat membuat rekomendasi KB
2	SRS – F – 15	Melihat rekomendasi KB	Akseptor dapat melihat hasil dari rekomendasi KB

4.1.2.2 Use Case Diagram Iterasi 1

Gambar 4.3 merupakan *use case diagram* untuk pengguna aplikasi KBku yang telah dilakukan evaluasi sebelumnya kepada pengguna KB dan bidan. Terdapat pengurangan kebutuhan pada beberapa fitur dan penambahan kebutuhan yang baru. Pengurangan dan penambahan kebutuhan tersebut digambarkan pada *use case diagram* iterasi ke 1. Untuk keseluruhan penggambaran *use case diagram* masih memiliki kesamaan dengan *use case diagram* yang sudah dibuat sebelumnya. Aplikasi KBku memiliki dua *actor* yaitu pengguna KB yang belum memiliki akun dan *actor* turunan dari pengguna KB ketika sudah memiliki akun di aplikasi KBku. Untuk pengguna KB belum dapat melakukan operasi pada fitur-fitur yang tersedia pada aplikasi KBku, namun ketika pengguna KB sudah mendaftarkan akun dan sudah dapat masuk ke dalam aplikasi, maka pengguna KB akan berubah menjadi akseptor. Akseptor disini dapat melakukan operasi pada fitur yang tersedia pada aplikasi KBku. Akseptor dapat melihat informasi KB, akseptor dapat membuat rekomendasi KB, akseptor dapat melihat informasi KB. Akseptor dapat melihat riwayat kalender KB. Sebelum melihat riwayat kalender KB, akseptor harus membuat kalender KB terlebih dahulu. Akseptor dapat menerima pemberitahuan KB dan dapat melihat isi dari pemberitahuan KB yang telah dibuat sebelumnya pada buat kalender KB. Akseptor dapat mencari rumah sakit terdekat. Akseptor dapat melihat profil, mengubah profil dan dapat keluar dari sistem.



A. Use Case Scenario untuk Akseptor

Berikut ini adalah *use case scenario* akseptor membuat rekomendasi KB. Penjelasan lengkap tentang akseptor menyimpan membuat KB akan dijelaskan pada Tabel 4.21.

Tabel 4.21 Use Case Scenario Akseptor Membuat Rekomendasi KB

41

	4. Akseptor memilih tombol hitung
<i>Alternative flows</i>	5. Jika ada data yang belum diisi maka sistem akan menampilkan pemberitahuan “isikan data dengan lengkap”
<i>Post-condition</i>	Akseptor dapat membuat rekomendasi KB

Pada Tabel 4.21 menjelaskan tentang akseptor membuat rekomendasi KB. Sebelum akseptor dapat membuat rekomendasi KB maka akseptor harus memiliki akun dan berhasil masuk kedalam aplikasi KBku. Untuk membuat rekomendasi KB, akseptor memilih menu rekomendasi KB pada halaman beranda, kemudian akseptor akan mengisi data yang sesuai dengan *form* yang telah disediakan pada aplikasi, kemudian memilih tombol hitung. Namun jika ada data yang belum diisi maka sistem akan menampilkan pemberitahuan untuk melengkapi isi data.

2. *Use Case Scenario* Akseptor Melihat Rekomendasi KB

Berikut ini adalah *use case scenario* akseptor melihat rekomendasi KB. Penjelasan lengkap tentang akseptor melihat rekomendasi KB akan dijelaskan pada Tabel 4.22.

Tabel 4.22 *Use Case Scenario* Akseptor Melihat Rekomendasi KB

<i>Objective</i>	Akseptor Melihat Rekomendasi KB
<i>Actor</i>	Akseptor
<i>Pre-condition</i>	Akseptor sudah memiliki akun pada aplikasi KBku, dan telah berhasil masuk kedalam aplikasi KBku
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Akseptor masuk ke halaman beranda 2. Akseptor memilih menu “Rekomendasi KB” 3. Akseptor mengisi data yang telah di sediakan pada <i>form</i> 4. Akseptor memilih tombol hitung 5. Sistem akan menampilkan hasil dari rekomendasi KB
<i>Alternative flows</i>	6. Jika ada data yang belum diisi maka sistem akan menampilkan pemberitahuan “isikan data dengan lengkap”
<i>Post-condition</i>	Akseptor dapat melihat hasil rekomendasi KB

Pada Tabel 4.22 menjelaskan tentang akseptor melihat rekomendasi KB. Sebelum akseptor dapat melihat rekomendasi KB maka akseptor harus memiliki akun dan berhasil masuk kedalam aplikasi KBku. Akseptor memilih menu rekomendasi KB pada halaman beranda, kemudian memasukkan data sesuai dengan *form* yang telah disediakan pada halaman buat rekomendasi KB, setelah

selesai mengisi *form*, akseptor memilih tombol hitung dan sistem akan menampilkan hasil dari rekomendasi KB. Namun jika ada data yang belum diisi maka sistem akan menampilkan pemberitahuan untuk melengkapi isi data.

3. *Use Case Scenario* Akseptor Membuat Kalender KB

Berikut ini adalah *use case scenario* akseptor membuat kalender KB. Penjelasan lengkap tentang akseptor membuat kalender KB dijelaskan pada Tabel 4.23.

Tabel 4.23 *Use Case Scenario* Akseptor Membuat Kalender KB

<i>Objective</i>	Akseptor Membuat Kalender KB
<i>Actor</i>	Akseptor
<i>Pre-condition</i>	Akseptor sudah memiliki akun pada aplikasi KBku, dan telah berhasil masuk kedalam aplikasi KBku
<i>Main flow</i>	<ol style="list-style-type: none"> 1. Akseptor memilih menu kalender KB pada halaman beranda 2. Akseptor memilih metode KB 3. Akseptor memasukkan tanggal KB terakhir 4. Akseptor memasukkan berat badan 5. Akseptor memasukkan tekanan darah 6. Akseptor memilih tombol buat pemberitahuan
<i>Alternative flows</i>	<ol style="list-style-type: none"> 7. Jika ada data yang belum diisi maka sistem akan menampilkan pemberitahuan “isikan data dengan lengkap”
<i>Post-condition</i>	Akseptor dapat membuat kalender KB

Pada Tabel 4.23 dijelaskan bahwa akseptor dapat membuat kalender KB. Untuk dapat membuat kalender KB, akseptor harus sudah memiliki akun dan berhasil masuk kedalam aplikasi KBku. Akseptor memilih menu kalender KB. Akseptor memilih metode KB yang digunakan, memasukkan jangka waktu pemberitahuan, tekanan darah terakhir serta berat badan terakhir dan memilih tombol buat pemberitahuan. Namun jika proses penyimpanan gagal maka akan muncul pemberitahuan.

4.1.3 Analisis Kebutuhan Iterasi 2

Analisis kebutuhan iterasi kedua merupakan analisis yang dilakukan ketika hasil dari analisis yang dilakukan pada iterasi pertama terdapat perubahan. Pada analisis kebutuhan iterasi kedua, dilakukan wawancara secara langsung untuk mengetahui apakah kebutuhan pada iterasi pertama sudah sesuai dengan apa yang pengguna KB inginkan. Pada iterasi kedua ini terdapat sedikit perubahan, yaitu pada saat melihat isi pemberitahuan KB, pengguna KB meminta ditambahkannya tombol untuk langsung menuju kehalaman mencari rumah sakit

terdekat. Analisis kebutuhan Iterasi ke 2 adalah iterasi terakhir, pengguna KB dan bidan sudah menyetujui kebutuhan yang telah dibuat oleh peneliti.

4.1.3.1 Analisis Kebutuhan Fungsional Iterasi 2

Tidak ada penambahan kebutuhan fungsional pada iterasi kedua. Hal ini dikarenakan pengguna KB hanya meminta penambahan tombol pada halaman isi dari pemberitahuan KB.

4.1.3.2 Use Case Diagram Iterasi 2

Gambar 4.4 merupakan *use case diagram* penggambaran dari hasil evaluasi pada iterasi ke dua. Pada gambar *usecase diagram* iterasi ke dua, hanya terdapat perubahan garis yang menghubungkan fitur melihat pemberitahuan KB dan mencari rumah sakit.



Gambar 4.4 Gambar Use Case Diagram Iterasi 2

BAB 5 PERANCANGAN DAN IMPEMENTASI

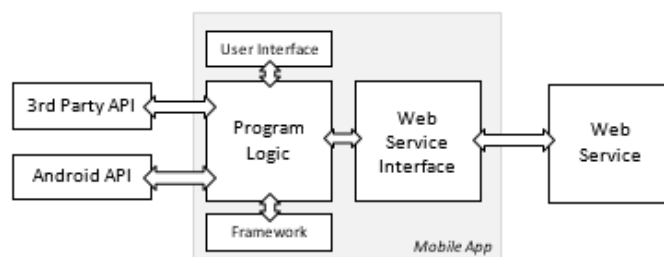
Pada bab lima ini membahas mengenai tahapan dari perancangan serta implementasi sistem aplikasi program KBku. Untuk perancangan dan implementasi sendiri menggunakan diagram alur proses UCD. Pertama adalah melakukan Perancangan Sistem, kedua adalah melakukan perancangan antarmuka dengan metode UCD, ketiga perancangan antarmuka yang sudah jadi atau *prototype* akan di evaluasi oleh pengguna, keempat melakukan analisis dan melakukan *update* jika perancangan antarmuka masih belum memberikan kepuasan dari sisi pengguna. Setelah perancangan selesai dilakukan, maka masuk ke tahap implementasi sistem. Pada tahapan implementasi sistem dijelaskan batasan-batasan implementasi, implementasi aksitekstur sistem, implementasi basis data, implementasi kode program.

5.1 Perancangan Sistem

Pada bagian perancangan sistem pada pembuatan aplikasi KBKU ini menggunakan metode UCD tahap *Produce design solution* dan *Evaluate design against user requirement* yang menerapkan perancangan antarmuka beserta iterasinya. Perancangan sistem berisi tentang alur kerja dari pengguna aplikasi, basis data, media yang mengakses aplikasi. Setelah tahap perancangan sistem, masuk ke tahap perancangan perangkat lunak, pada tahapan perangkat lunak terdiri dari beberapa alur kerja dari perancangan arsitektur sistem, perancangan basis data, perancangan *class diagram*, perancangan *sequence diagram*, perancangan algoritme dan perancangan antarmuka.

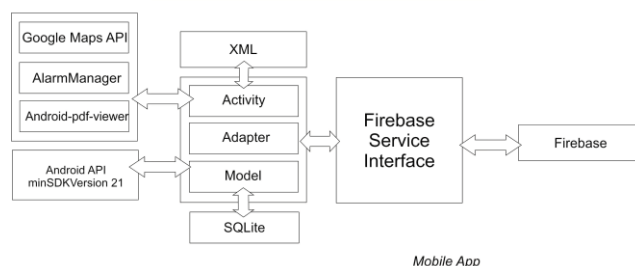
5.1.1 Perancangan Arsitektur Sistem

Perancangan arsitektur aplikasi KBku ini dibangun menggunakan arsitektur *front-end* dan *back-end* yang memisahkan antara pengolahan antarmuka sistem dan pengolahan data yang ada dalam arsitektur *back-end*. Konsep *mobile application* pada perangkat *mobile* dijalankan dengan sistem operasi Android. Konsep aplikasi ini dibangun menggunakan Bahasa Java serta XML untuk tampilan aplikasinya. Berikut ini merupakan rancangan arsitektur sistem aplikasi KBku dapat dilihat pada gambar 5.2 dan 5.3.



Gambar 5.1 Arsitektur Sistem Aplikasi Mobile

Sumber: (STAPIC, 2013)



Gambar 5.2 Arsitektur Sistem Aplikasi KBku

Arsitektur sistem pada bagian *mobile app* memiliki empat bagian. Empat bagian tersebut adalah *user interface*, *program logic*, *web service interface*, dan *local database*. Pada aplikasi KBku keempat bagian tersebut digunakan untuk:

1. *User Interface* : Berisi berkas XML yang digunakan untuk mengatur tampilan *activity* aplikasi
2. *Program Logic* : Berisi kumpulan *java class* yang terdiri dari *class activity*, *model*, dan *adapter*
3. *Local Database* : Terdapat SQLite yang digunakan dalam aplikasi yang dikembangkan
4. *Web Service Interface* : berisi *firebase service interface* yang digunakan untuk menjembatani antara *mobile app* dan *firebase* itu sendiri.

Android API dan 3rd Party API juga digunakan dalam aplikasi yang sedang dikembangkan.

1. Android API menerapkan minSDKVersion 21 atau setara dengan OS Android Lollipop keatas.
2. 3rd Party API terdiri dari:
 - a. Google Maps API : digunakan untuk mencari rumah sakit terdekat
 - b. AlarmManager : digunakan untuk memberikan pemberitahuan waktu KB
 - c. Android-pdf-viewer: digunakan untuk menampilkan berkas *.pdf

5.1.2 Perancangan Basis Data

Perancangan basis data pada aplikasi KBku menggunakan skema diagram, hal ini dikarenakan aplikasi KBku menggunakan basis data *Firebase Real-time database*, sehingga perancangan basis data yang dibuat tidak memiliki relasi. Untuk Gambar dari perancangan basis data aplikasi KBku dapat dilihat pada Gambar 5.4.

5.1.2.1 Skema Basis Data RiwayatKB

Berikut merupakan skema basis data dari riwayat KB. Skema basis data riwayat KB digunakan untuk menyimpan data pada halaman riwayat KB. Isi dari riwayat KB bertipe *string*.

```

"RiwayatKB" : {
  "type": "object",

```

```

    "properties": {
      "Id": {
        "type": "object",
        "properties": {
          "idRiwayatKB": {
            "type": "String"
          }, "Sistol" : {
            "type": "String"
          }, "diastol" : {
            "type": "String"
          }, "beratbadan" : {
            "type": "String"
          }
        }
      }
    }
  }
}

```

Gambar 5.3 Gambar Perancangan Basis Data Riwayat KB

Berikut merupakan keterangan dari isi skema basis data riwayat KB, yang bisa dilihat pada Tabel 5.1.

Tabel 5.1 Keterangan Skema Basis Data Riwayat KB

No	Nama Field	Tipe Data	Keterangan
1	idRiwayat	String	Untuk menyimpan id riwayat KB
2	sistol	String	Untuk menyimpan data sistol
3	diastol	String	Untuk menyimpan data diastol
4	beratbadan	String	Untuk menyimpan data berat badan

5.1.2.2 Skema Basis Data PemberitahuanKB

Berikut merupakan skema basis data dari pemberitahuan KB. Skema basis data pemberitahuan KB digunakan untuk menyimpan data pada pemberitahuan KB. Isi dari pemberitahuan KB bertipe string.

```

"PemberitahuanKB" : {
  "type" : "object"
  "properties" : {
    "id":{
      "type": "object"
      "properties":
    } "idPemberitahuanKB":{
      "type": "String"
    }
  }
}

```

```

    }, "tanggalTerakhir":{
      "type":"String"
    }, "metode":{
      "type":"String"
    }
  }
}

```

Gambar 5.4 Gambar Perancangan Basis Data Pemberitahuan KB

Berikut merupakan keterangan dari isi skema basis data pemberitahuan KB, yang bisa dilihat pada Tabel 5.2.

Tabel 5.2 Keterangan Skema Basis Data Pemberitahuan KB

No	Nama Field	Tipe Data	Keterangan
1	IdPemberitahuanKB	String	Untuk menyimpan id pemberitahuan KB
2	tanggalAkhir	String	Untuk menyimpan data tanggal akhir melakukan KB
3	metode	String	Untuk menyimpan data metode kontrasepsi yang digunakan

5.1.2.3 Skema Basis Data Akseptor

Berikut merupakan skema basis data dari data akseptor. Skema basis data akseptor digunakan untuk menyimpan data akseptor ketika melakukan ubah profil. Isi dari riwayat KB bertipe string.

```

"Akseptor" : {
  "type":"object"
  "properties":{
    "id":{
      "type":"object"
      "properties":
      { "idAkseptor":{
        "type":"String"
      }, "Alamat":{
        "type":"String"
      }, "Nama":{
        "type":"String"
      }, "NoHP":"String"
      }
    }
  }
}

```

Gambar 5.5 Gambar Perancangan Basis Data Akseptor

Berikut merupakan keterangan dari isi skema basis data akseptor, yang bisa dilihat pada Tabel 5.3.

Tabel 5.3 Keterangan Skema Basis Data Akseptor

No	Nama <i>Field</i>	Tipe Data	Keterangan
1	idAkseptor	<i>String</i>	Untuk menyimpan id akseptor
2	Alamat	<i>String</i>	Untuk menyimpan data alamat
3	Nama	<i>String</i>	Untuk menyimpan data nama
4	NoHP	<i>String</i>	Untuk menyimpan data nomor <i>handphone</i>

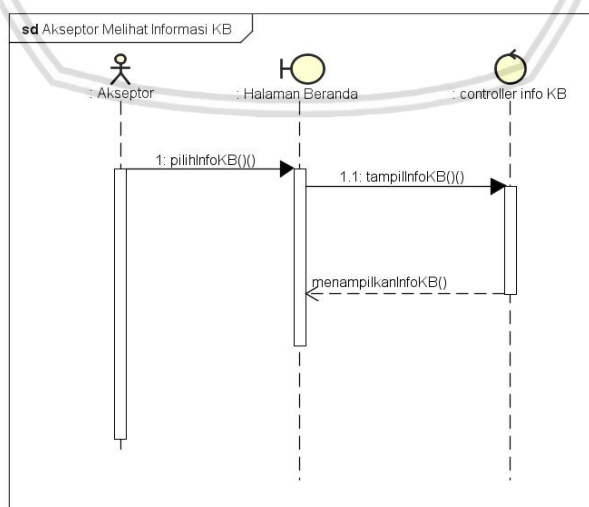
5.1.3 Perancangan *Sequence Diagram*

Perancangan *sequence diagram* digunakan untuk mendeskripsikan bagaimana dan pesan yang di sampaikan oleh sebuah *entitas* dalam sistem melakukan interaksi. Aplikasi KBku sendiri memiliki beberapa interaksi pengguna terhadap sistem. Interaksi tersebut dapat dilihat pada Gambar *sequence diagram* berikut ini.

A. *Sequence Diagram* untuk Akseptor

1. *Sequence Diagram* Akseptor Melihat Informasi KB

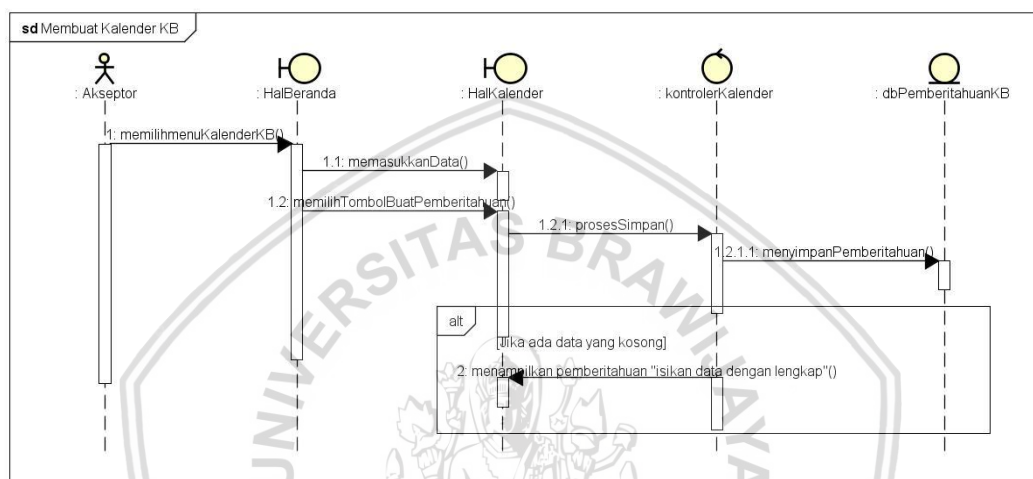
Sequence diagram pada Gambar 5.5 menjelaskan bahwa yang menjadi aktor adalah akseptor. Ketika akseptor melihat informasi KB, maka akseptor akan melakukan aksi yang pertama adalah memilih menu *infoKB* pada halaman beranda. Kemudian sistem akan menampilkan halaman *informasiKB* dengan bantuan *controller info KB*.



Gambar 5.6 Gambar *Sequence Diagram* Akseptor Melihat Informasi KB

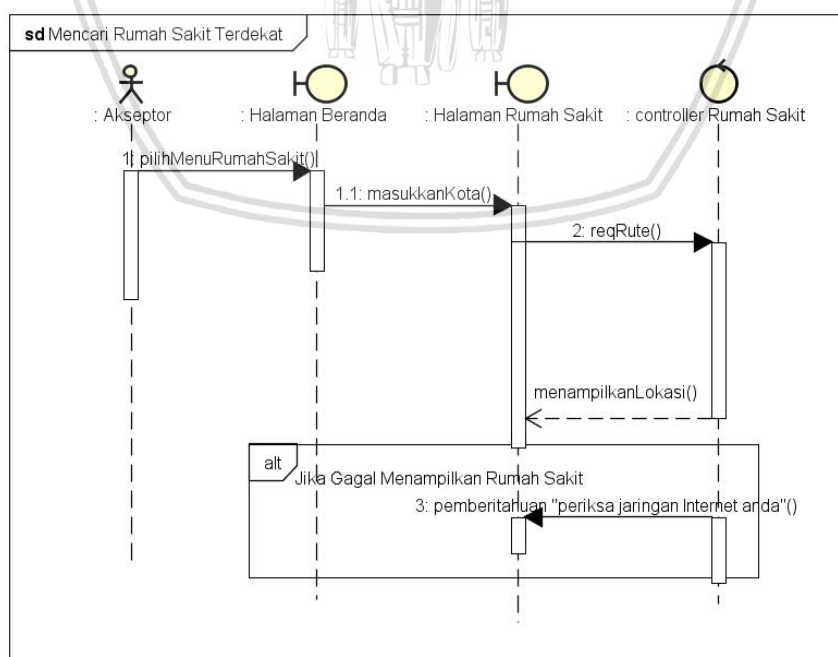
2. Sequence Diagram Akseptor Membuat Kalender KB

Sequence diagram pada Gambar 5.6 menjelaskan bahwa yang menjadi aktor adalah akseptor. Akseptor memilih menu kalender KB pada halaman beranda, kemudian akseptor diminta memasukkan data dan memilih tombol buat pemberitahuan untuk menyimpan data yang sudah dibuat. Data yang sudah dibuat akan disimpan dengan menggunakan bantuan controller kalender, jika data berhasil disimpan, maka data akan disimpan pada dbPemberitahuanKB. Namun jika ada data yang masih kosong, maka akan muncul pemberitahuan gagal dan akseptor diminta untuk melengkapi data yang kosong tersebut.



Gambar 5.7 Gambar Sequence Diagram Akseptor Melihat Informasi KB

3. Sequence Diagram Akseptor Mencari Rumah Sakit



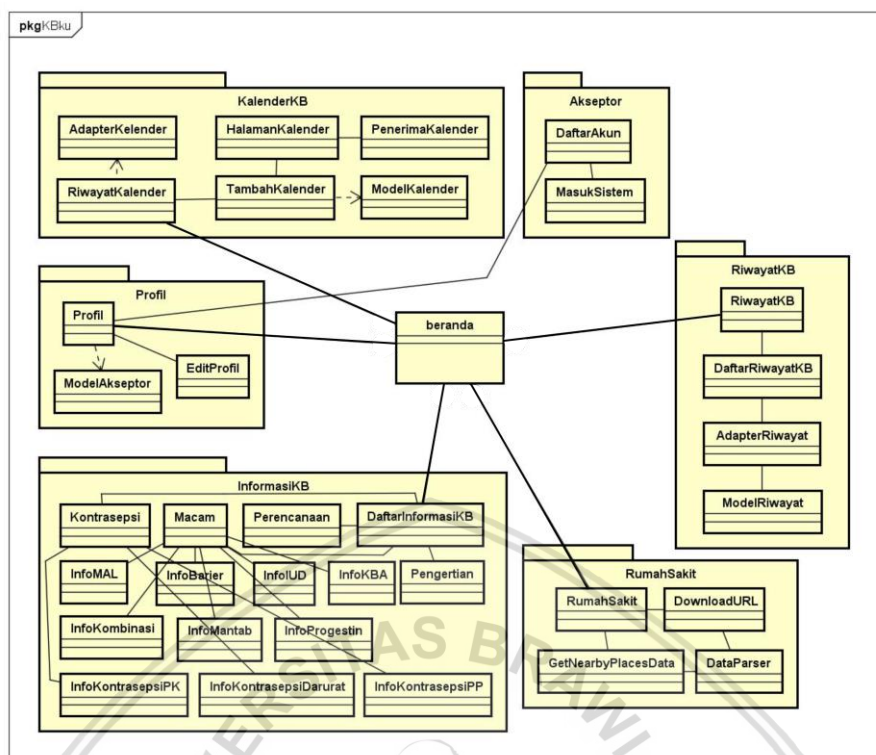
Gambar 5.8 Gambar Sequence Diagram Akseptor Pemberitahuan KB

Sequence diagram pada Gambar 5.7 menjelaskan bahwa yang menjadi aktor adalah akseptor. Memilih menu rumah sakit pada halaman beranda, kemudian pada halaman beranda akseptor memasukkan lokasi yang ingin diketahui rutanya, kemudian sistem akan menampilkan rute dengan menggunakan bantuan *controller* rumah sakit. Jika gagal maka akan muncul pemberitahuan periksa jaringan *internet* anda.

5.1.4 Perancangan *Class Diagram*

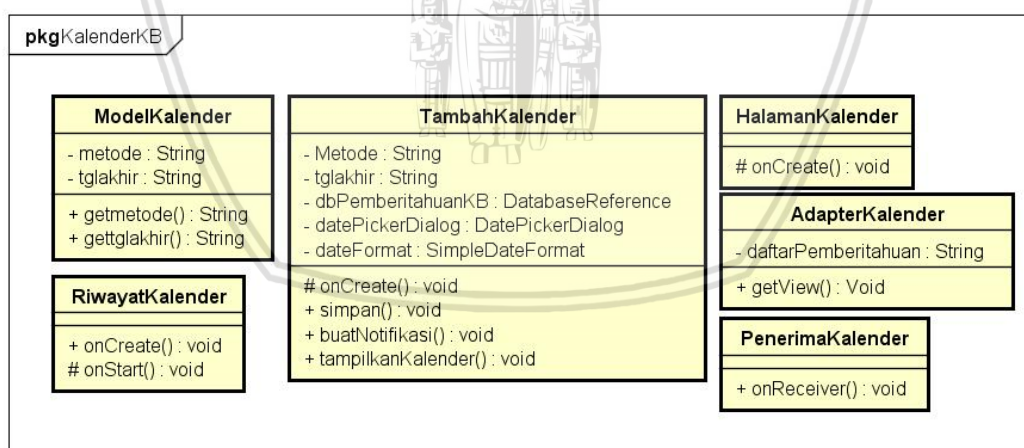
Perancangan *class diagram* merupakan proses perancangan yang digunakan untuk menunjukkan keterhubungan sebuah *class* dengan sistem. Dalam membangun aplikasi KBku, *class* dikelompokkan berdasarkan fungsi yang akan diimplementasikan kedalam sistem. Berikut adalah *class diagram* aplikasi KBku. Pada Gambar 5.7 merupakan perancangan *class diagram* pada aplikasi KBku. Pada *class diagram* tersebut dijelaskan bahwa aplikasi KBku memiliki enam paket. Didalam masing-masing paket terdiri dari beberapa *class*. Terdapat dua paket yang memiliki *class* model dan satu *class* adapter. *Class* model dan *class* adapter ini digunakan untuk menghubungkan isi data yang akan ditampilkan atau akan diolah oleh *firebase database*. *Class* yang memiliki *class* model dan *class* adapter adalah paket RiwayatKB dan KalenderKB. Pada paket RiwayatKB memiliki dua *class* yang terhubung dengan *class* modelRiwayat, yaitu RiwayatKB, DaftarRiwayatKB, dan AdapterRiwayat. Kemudian paket lain yang memiliki *class* model dan *class* adapter pada paketnya adalah KalenderKB. Paket KalenderKB terdiri dari empat *class* yang digunakan untuk menampilkan, menyimpan, dan menerima kalenderKB, dan *class* adapterKalender yang digunakan untuk menampilkan data pada *listView*, serta *class* ModelKalender yang digunakan untuk menginisialisasi data yang berhubungan dengan penyimpanan *firebase database*.

Paket lainnya adalah InformasiKB, Profil, Akseptor, dan RumahSakit. Pada keempat paket ini berisi *class-class* yang digunakan untuk menampilkan sebuah data yang tidak memerlukan *class* model dan *class* adapter. Semua paket pada aplikasi KBku terhubung pada satu *class* utama yaitu *class* beranda.



Gambar 5.9 Gambar Perancangan *Class Diagram* Aplikasi KBku

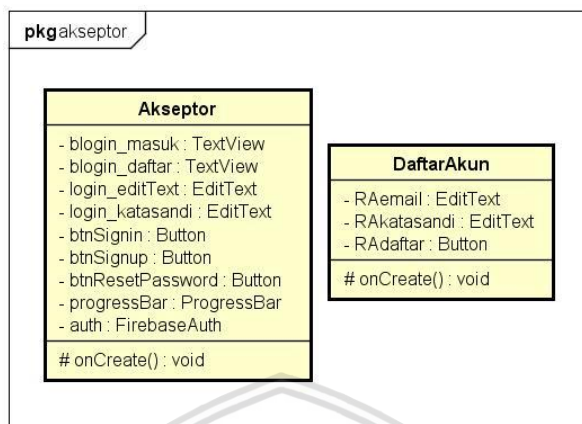
5.1.4.1 Perancangan *Class Diagram* Paket KalenderKB



Gambar 5.10 Gambar Perancangan *Class Diagram* Paket Kalender KB

Berdasarkan Gambar 5.10 Paket KalenderKB memiliki enam kelas, yaitu ModelKalender, RiwayatKalender, HalamanKalender, AdapterKalender, PenerimaKalender, dan TambahKalender. Kelas TambahKalender merupakan *mainclass* dari paket KalenderKB. KalenderKB sendiri digunakan pada fitur KalenderKB yang berfungsi untuk membuat pemberitahuan KB.

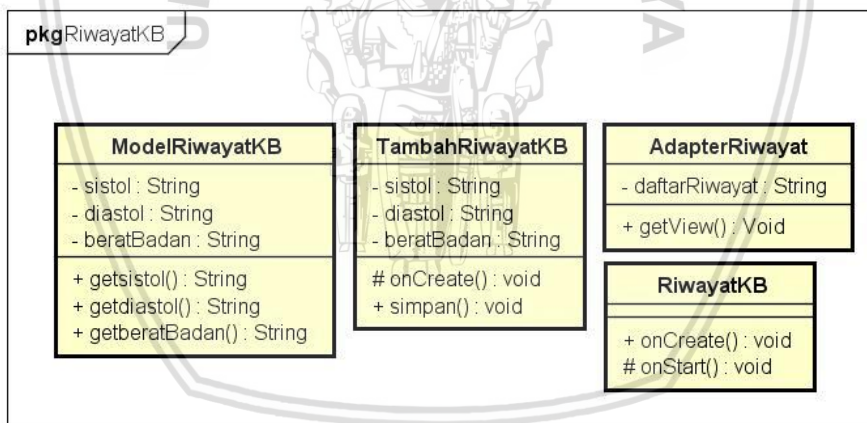
5.1.4.2 Perancangan Class Diagram Paket Akseptor



Gambar 5.11 Gambar Perancangan Class Diagram Paket Akseptor

Berdasarkan Gambar 5.11 Paket Akseptor memiliki dua kelas, yaitu Akseptor dan DaftarAkun. Kelas Akseptor digunakan pada fitur masuk kedalam sistem, sedangkan DaftarAkun digunakan untuk fitur daftar akun.

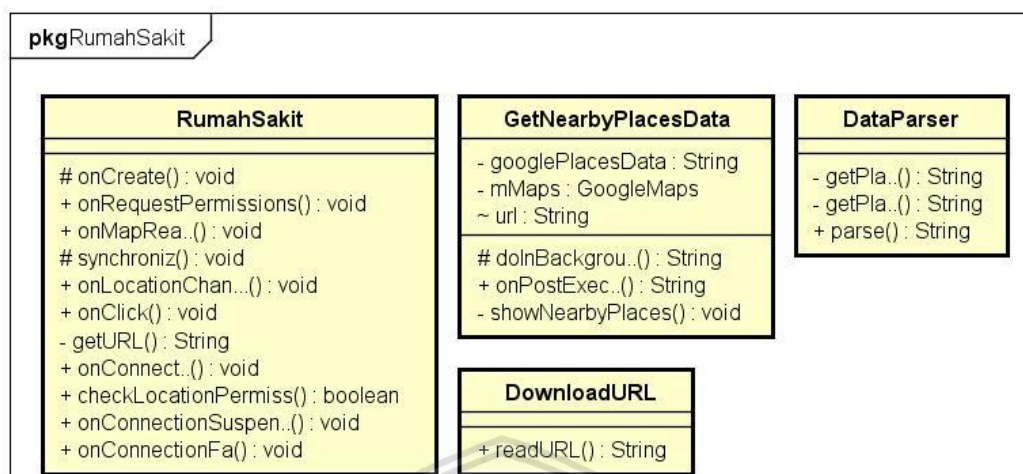
5.1.4.3 Perancangan Class Diagram Paket Riwayat KB



Gambar 5.12 Gambar Perancangan Class Diagram Paket Riwayat KB

Berdasarkan Gambar 5.12 Paket RiwayatKB memiliki empat kelas, yaitu kelas ModelRiwayatKB, TambahRiwayatKB, AdapterRiwayat, dan RiwayatKB. Kelas utama pada paket RiwayatKB adalah TambahRiwayatKB. Paket RiwayatKB ini digunakan pada fitur riwayat KB, sehingga dalam fitur ini dapat dilakukan pembuatan dan penyimpanan riwayat KB yang terdiri dari tekanan darah yaitu sistol, diastole dan beratbadan.

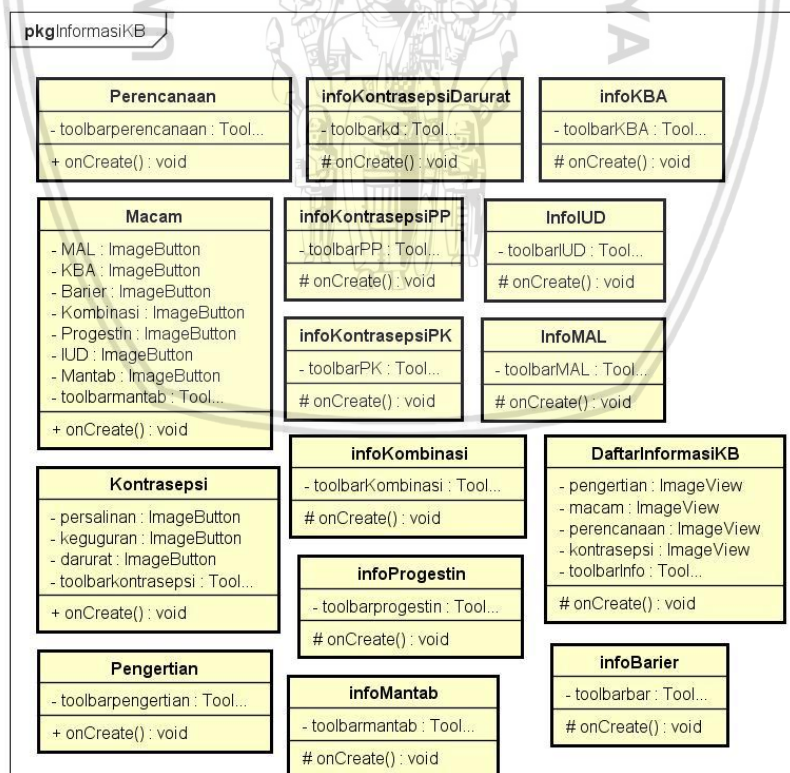
5.1.4.4 Perancangan Class Diagram Paket RumahSakit



Gambar 5.13 Gambar Perancangan Class Diagram Paket Rumah Sakit

Berdasarkan Gambar 5.13 paket RumahSakit memiliki empat kelas, yaitu RumahSakit, GetNearbyData, DownloadURL, DataParser. Paket RumahSakit ini digunakan pada fitur mencari rumah sakit terdekat.

5.1.4.5 Perancangan Class Diagram Paket InformasiKB

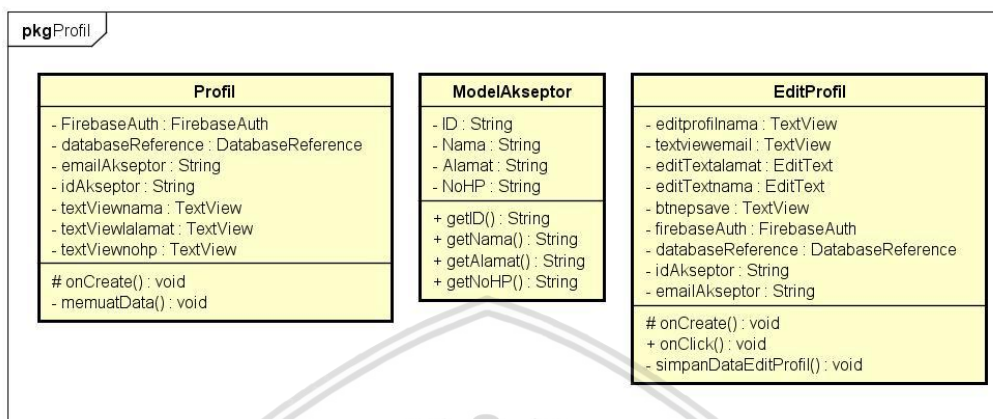


Gambar 5.14 Gambar Perancangan Class Diagram Paket Informasi KB

Berdasarkan Gambar 5.14 paket InformasiKB memiliki lima belas kelas, dengan *mainclass* DaftarInformasiKB. Paket InformasiKB ini digunakan

pada fitur informasi KB, didalam kelas-kelas pada paket InformasiKB terdapat informasi yang disimpan dalam bentuk .pdf yang berisi tentang macam-macam informasi KB.

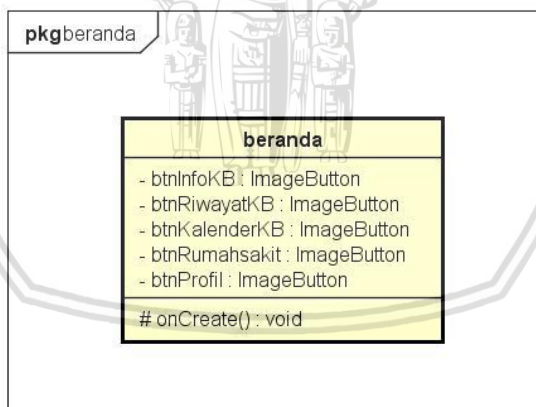
5.1.4.6 Perancangan Class Diagram Paket Profil



Gambar 5.15 Gambar Perancangan Class Diagram Paket Profil

Berdasarkan Gambar 5.15 paket Profil memiliki tiga kelas, yaitu Profil, ModelAkteptor, EditProfil. Pada paket Profil kelas Profil digunakan untuk fitur melihat profil dan kelas EditProfil digunakan untuk mengubah profil.

5.1.4.7 Perancangan Class Diagram Beranda



Gambar 5.16 Gambar Perancangan Class Diagram Beranda

Berdasarkan Gambar 5.16 paket beranda hanya memiliki satu kelas, satu kelas tersebut digunakan juga sebagai *mainclass* pada *class diagram* KBku.

5.1.5 Perancangan Antarmuka

Pada perancangan antarmuka digunakan agar pengguna KB memiliki gambaran tentang antarmuka aplikasi yang nantinya digunakan oleh pengguna KB. Berikut merupakan perancangan antarmuka yang nantinya akan diimplementasikan kedalam aplikasi KBku.

1. Halaman Masuk ke Sistem

Keterangan:

1. Kolom memasukkan email
2. Kolom memasukkan kata sandi
3. Tombol daftar
4. Tombol masuk
5. Logo KBku

Gambar 5.17 Gambar Perancangan Antarmuka Halaman Masuk ke Sistem

Pada Gambar 5.17 merupakan Gambaran tentang perancangan antarmuka masuk kedalam sistem. Agar pengguna KB dapat masuk kedalam sistem, maka pengguna KB diharuskan mengisi form masuk ke sistem yang berisi alamat *email* serta kata sandi dan memilih tombol masuk. Namun jika pengguna KB belum memiliki akun, maka pengguna KB diharuskan memilih tombol daftar untuk mendaftarkan akun.

2. Halaman Daftar Akun

Keterangan:

1. Logo KBku
2. Kata-kata untuk mensukseskan program KB
3. Kolom untuk mendaftarkan email
4. Kolom untuk mendaftarkan kata sandi
5. Tombol daftar

Gambar 5.18 Gambar Perancangan Antarmuka Halaman Daftar Akun

Pada Gambar 5.18 merupakan Gambaran tentang perancangan antarmuka untuk mendaftar akun. Pada perancangan antarmuka mendaftar akun terdiri dari logo KBku pada bagian atas halaman, dan dibawah Gambar logo KB terdapat kata-kata yang berisi ajakan untuk menggunakan KB. Kemudian pengguna KB dapat mengisi data untuk mendaftarkan akun sesuai dengan form yang telah disediakan. Setelah selesai mengisi form yang telah disediakan pengguna KB memilih tombol daftar.

3. Halaman Beranda

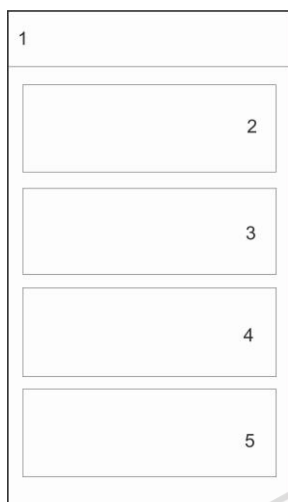
Pada Gambar 5.19 menjelaskan tentang perancangan antarmuka halaman beranda. Pada halaman beranda ini terdapat logo aplikasi KBku pada bagian atas dan terdapat lima menu yaitu informasi KB, riwayat KB, Kalender KB, Rumah sakit, serta Profil. Didalam menu tersebut terdapat *icon* dan judul menu.



Gambar 5.19 Gambar Perancangan Antarmuka Halaman Beranda

4. Halaman Daftar Informasi KB

Pada Gambar 5.20 merupakan halaman daftar informasi KB. Pada halaman daftar informasi KB ini terdiri dari beberapa informasi KB yang bisa dipilih oleh akseptor sesuai dengan berita yang diinginkan. Pada halaman informasi KB memiliki menu yang berisi Gambar serta judul dari berita.



Keterangan:

1. Judul halaman Informasi KB
2. Menu Pengertian KB
3. Menu Macam-macam KB
4. Menu Perencanaan KB
5. Menu Kontrasepsi

Gambar 5.20 Gambar Perancangan Antarmuka Halaman Daftar Informasi KB

5. Halaman Detail Informasi KB

Pada Gambar 5.21 merupakan halaman detail informasi KB. Pada halaman detail informasi KB terdiri dari judul halaman informasi KB dan isi informasi KB.

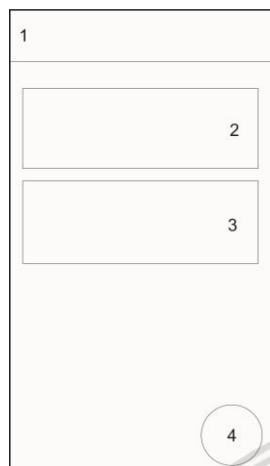


Keterangan:

1. Judul Halaman Informasi KB
2. Isi Informasi KB

Gambar 5.21 Gambar Perancangan Antarmuka Halaman Detail Informasi KB

6. Halaman Daftar Riwayat KB



Keterangan:

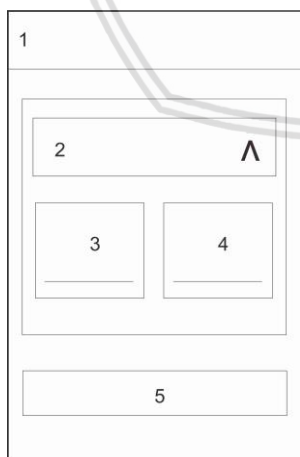
1. Judul halaman riwayat KB
2. Daftar riwayat KB yang sudah pernah dibuat
3. Daftar riwayat KB yang sudah pernah dibuat
4. Tombol untuk buat riwayat KB baru

Gambar 5.22 Gambar Perancangan Antarmuka Halaman Daftar Riwayat KB

Gambar 5.22 merupakan perancangan halaman daftar riwayat KB, pada halaman ini terdiri dari judul halaman, dan daftar riwayat KB yang telah dibuat sebelumnya serta tombol untuk menambah riwayat KB baru.

7. Halaman Buat Riwayat KB

Gambar 5.23 merupakan perancangan antarmuka halaman buat riwayat KB. Pada halaman ini terdapat judul halaman, pilihan metode, kolom untuk memasukkan tekanan darah dan tombol simpan.

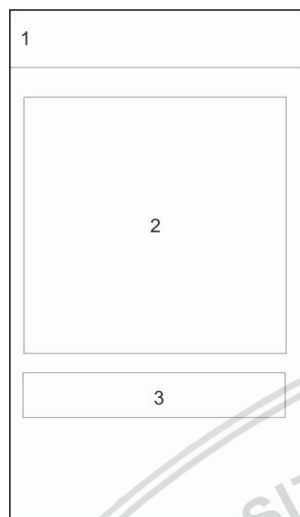


Keterangan:

1. Judul halaman buat riwayat KB
2. Pilihan metode KB
3. Memasukkan tekanan darah sistol
4. Memasukkan tekanan darah diastole
5. Tombol simpan

Gambar 5.23 Gambar Perancangan Antarmuka Halaman Buat Riwayat KB

8. Halaman Detail Riwayat KB



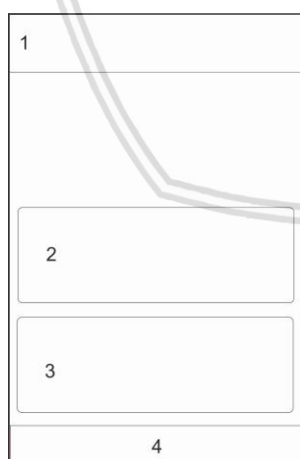
Keterangan :

1. Judul Halaman isi riwayat KB
2. Isi dari riwayat KB yang sudah dibuat
3. Tombol ok

Gambar 5.24 Gambar Perancangan Antarmuka Halaman Detail Riwayat KB

Gambar 5.24 merupakan hasil dari perancangan detail riwayat KB. Pada halaman detail riwayat KB berisi tentang judul halaman riwayat KB, isi dari riwayat KB yang telah dibuat sebelumnya dan tombol ok.

9. Daftar Kalender KB



Keterangan :

1. Judul halaman daftar kalender KB
2. Daftar kalender KB yang pernah dilakukan
3. Daftar kalender KB yang pernah dilakukan
4. Tombol buat kalender baru

Gambar 5.25 Gambar Perancangan Antarmuka Halaman Daftar Kalender

Gambar 5.25 merupakan gambar perancangan antarmuka daftar kalender yang telah dibuat sebelumnya. Pada halaman ini terdiri dari judul halaman dan daftar kalender yang dibuat sebelumnya serta tombol untuk membuat kalender baru.

10. Halaman Buat Kalender KB

Keterangan :

1. Judul halaman buat kalender KB
2. Pilih metode KB
3. Memasukkan tanggal terakhir KB
4. Tombol buat pemberitahuan

Gambar 5.26 Gambar Perancangan Antarmuka Halaman Buat Kalender KB

Gambar 5.26 merupakan perancangan halaman buat kalender KB, pada halaman ini terdiri dari judul halaman pemberitahuan KB, pemilihan metode yang digunakan dan tanggal terakhir melakukan program KB, serta tombol untuk menyimpan data buat pemberitahuan KB.

11. Halaman Isi Pemberitahuan KB

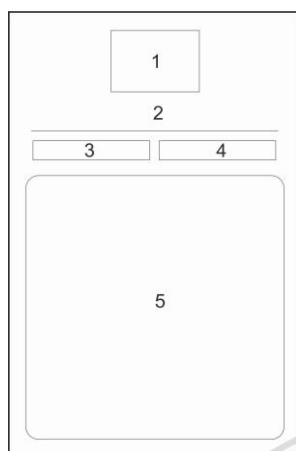
Gambar 5.27 merupakan perancangan untuk halaman pemberitahuan KB. Pada halaman ini terdiri dari logo KBku, kemudian isi dari pemberitahuan KB dan tombol ok yang akan menghubungkan kembali ke halaman beranda.

Keterangan :

1. Logo KBku
2. Isi dari pemberitahuan KB
3. Tombol ok

Gambar 5.27 Gambar Perancangan Antarmuka Halaman Isi Pemberitahuan KB

12. Halaman Rumah sakit



Keterangan :

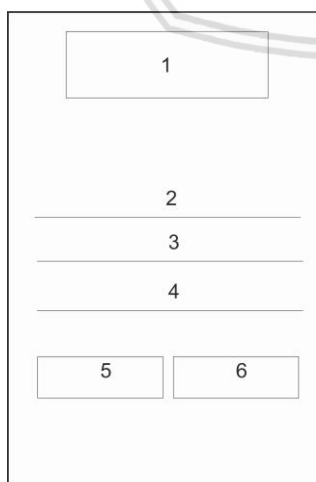
1. Logo KBku
2. Kolom untuk memasukkan tempat saat ini
3. Tombol untuk mencari kota
4. Tombol untuk menampilkan rumah sakit terdekat
5. tampilan *maps* dari rumah sakit terdekat

Gambar 5.28 Gambar Perancangan Antarmuka Halaman Mencari Rumah sakit

Gambar 2.28 merupakan gambar perancangan sistem untuk mencari rumah sakit terdekat. Pada halaman ini terdiri dari logo KBku, kolom untuk memasukkan kota yang ingin dicari, kemudian tombol untuk mencari kota saat ini dan tombol mencari rumah sakit terdekat. Dan pada bagian bawah tombol terdapat peta yang akan menampilkan rumah sakit terdekat dan rute untuk menuju rumah sakit yang dituju.

13. Halaman Profil

Pada Gambar 5.29 merupakan perancangan halaman profil, pada bagian atas halaman profil terdapat logo KBku. Kemudian terdapat isi dari profil pengguna. Pada bagian bawah halaman terdapat dua tombol yaitu tombol edit profil dan tombol keluar sistem.



Keterangan :

1. Logo KBku
2. Tampilan Nama
3. Tampilan Alamat
4. Tampilan No. Hp
5. Tombol ubah profil
6. Tombol keluar dari aplikasi KBku

Gambar 5.29 Gambar Perancangan Antarmuka Halaman Profil

14. Halaman Ubah Profil

Gambar 2.30 merupakan Gambar perancangan halaman edit profil. Pada bagian atas halaman ubah profil terdapat logo aplikasi KBku. Kemudian pada bagian tengah halaman terdapat form yang digunakan untuk mengisi data yang akan disimpan dan akan ditampilkan pada profil. Selanjutnya terdapat tombol simpan yang digunakan untuk menyimpan data yang telah diisikan.



The diagram shows a vertical form layout. At the top is a rectangular box labeled '1'. Below it is a horizontal line labeled '2'. This is followed by three more horizontal lines labeled '3', '4', and '5' respectively. At the bottom is a rectangular box labeled '6'.

Keterangan :	
1.	Logo KBku
2.	Tampilan alamat email
3.	Kolom untuk merubah nama
4.	Kolom untuk merubah alamat
5.	Kolom untuk merubah no.hp
6.	Tombol simpan

Gambar 5.30 Gambar Perancangan Antarmuka Halaman Edit Profil

5.2 Perancangan Sistem Iterasi 1

Merujuk pada iterasi yang terdapat pada analisis kebutuhan, dimana fitur riwayat KB dihapus dan dari fitur riwayat KB dijadikan satu dengan Pemberitahuan KB maka perancangan basis data pemberitahuan KB berubah. Dan juga adanya penambahan fitur rekomendasi KB, maka ditambahkan database baru untuk fitur rekomendasi KB.

5.2.1 Perancangan Basis Data Iterasi 1

Perubahan pada perancangan basis data iterasi pertama ini, merujuk pada analisa kebutuhan pada iterasi pertama, kebutuhan riwayat KB digabung menjadi satu dengan kalender KB, maka perancangan basis data riwayat KB dihapus dan perancangan basis data pemberitahuan KB berubah. Dan juga penambahan perancangan basis data untuk rekomendasi KB.

5.2.1.1 Skema Basis Data PemberitahuanKB Iterasi 1

Berikut merupakan skema basis data dari pemberitahuan KB. Skema basis data pemberitahuan KB digunakan untuk menyimpan data pada pemberitahuan KB. Isi dari pemberitahuan KB bertipe `string`.

```
"PemberitahuanKB" : {
  "type": "object",
  "properties": {
```

```

    "Id": {
      "type": "object",
      "properties": {
        "idPemberitahuanKB": {
          "type": "String"
        }, "Metode": {
          "type": "String"
        }, "Sistol" : {
          "type": "String"
        }, "diastol" : {
          "type": "String"
        }, "beratbadan" : {
          "type": "String"
        }
      }
    }
  }
}

```

Gambar 5.31 Gambar Perancangan Basis Data Pemberitahuan KB Iterasi 1

Berikut merupakan keterangan dari isi skema basis data Pemberitahuan KB, yang bisa dilihat pada Tabel 5.4

Tabel 5.4 Keterangan Skema Basis Data Pemberitahuan KB Iterasi 1

No	Nama Field	Tipe Data	Keterangan
1	idPemberitahuanKB	String	Untuk menyimpan id pemberitahuan KB
2	Metode	String	Untuk menyimpan data metode
3	Sistol	String	Untuk menyimpan data sistol
4	diastol	String	Untuk menyimpan data diastole
5	beratbadan	String	Untuk menyimpan data berat badan

5.2.1.2 Skema Basis Data RekomendasiKB

Berbeda dengan perancangan basis data sebelumnya, perancangan basis data pada rekomendasi KB tidak menggunakan *firebase*, namun menggunakan basis data SQL. Basis data SQL sendiri berisifat statis dan dapat menampung jumlah data yang banyak maka sangat cocok untuk menyimpan data rekomendasi KB. ER Diagram dari rekomendasiKB dapat dilihat pada Gambar 5.32.

rekomendasiKB

id
v1
v2
v3
v4
v5
v6
v7
v8
v9
kelas

Gambar 5.32 Gambar Perancangan Basis Data Rekomendasi KB

Keterangan maksud dari v1 hingga v9 :

- V1 : Umur
 V2 : Pendidikan Istri
 V3 : Pendidikan Suami
 V4 : Jumlah anak
 V5 : Agama Istri
 V6 : Status kerja istri
 V7 : Kesibukan Suami
 V8 : Standar kehidupan
 V9 : Informasi Kontrasepsi

Berikut merupakan keterangan dari isi skema basis data Rekomendasi KB, yang bisa dilihat pada Tabel 5.5.

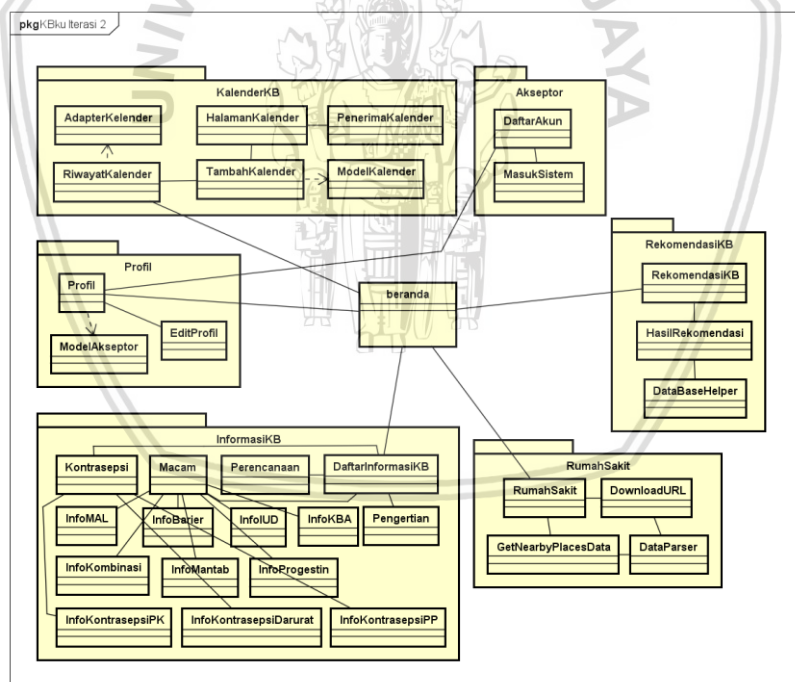
Tabel 5.5 Keterangan Skema Basis Data Rekomendasi KB

No	Nama Field	Tipe Data	Keterangan
1	Id	INTEGER	Untuk menyimpan id rekomendasi KB
2	v1	INTEGER	Kolom umur istri untuk menyimpan data umur istri
3	v2	INTEGER	Pilihan jawaban untuk menyimpan data pendidikan istri
4	v3	INTEGER	Pilihan jawaban untuk pendidikan suami
5	v4	INTEGER	Kolom jumlah anak untuk menyimpan data jumlah anak
6	v5	INTEGER	Pilihan jawaban untuk agama istri

7	v6	INTEGER	Pilihan jawaban untuk menyimpan data status kerja
8	v7	INTEGER	Pilihan jawaban untuk menyimpan data kesibukan suami
9	v8	INTEGER	Pilihan jawaban untuk menyimpan data standar kehidupan
10	v9	INTEGER	Pilihan jawaban untuk menyimpan data informasi kontrasepsi
11	kelas	INTEGER	Untuk menyimpan hasil dari kelas rekomendasi KB

5.2.2 Perancangan *Class Diagram Iterasi 1*

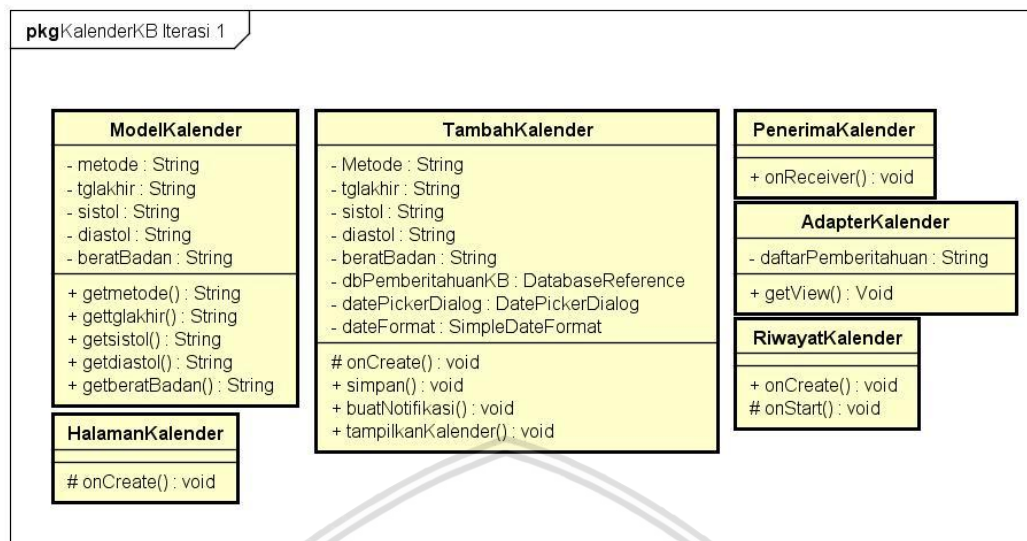
Merujuk pada perubahan analisis kebutuhan yang menghasilkan keputusan bahwa fitur riwayat KB dihapus dan di tambahkannya fitur rekomendasi KB, maka perancangan *class diagram* berubah. Untuk perancangan *class diagram* iterasi 1 dapat dilihat pada Gambar 5.33.



Gambar 5.33 Gambar Perancangan *Class Diagram* Aplikasi KBku Iterasi 1

Berdasarkan Gambar 5.33 tidak ada perubahan yang banyak pada perancangan *class diagram* pada iterasi 1 ini. Hanya saja paket RiwayatKB dihapus dan diganti oleh paket RekomendasiKB.

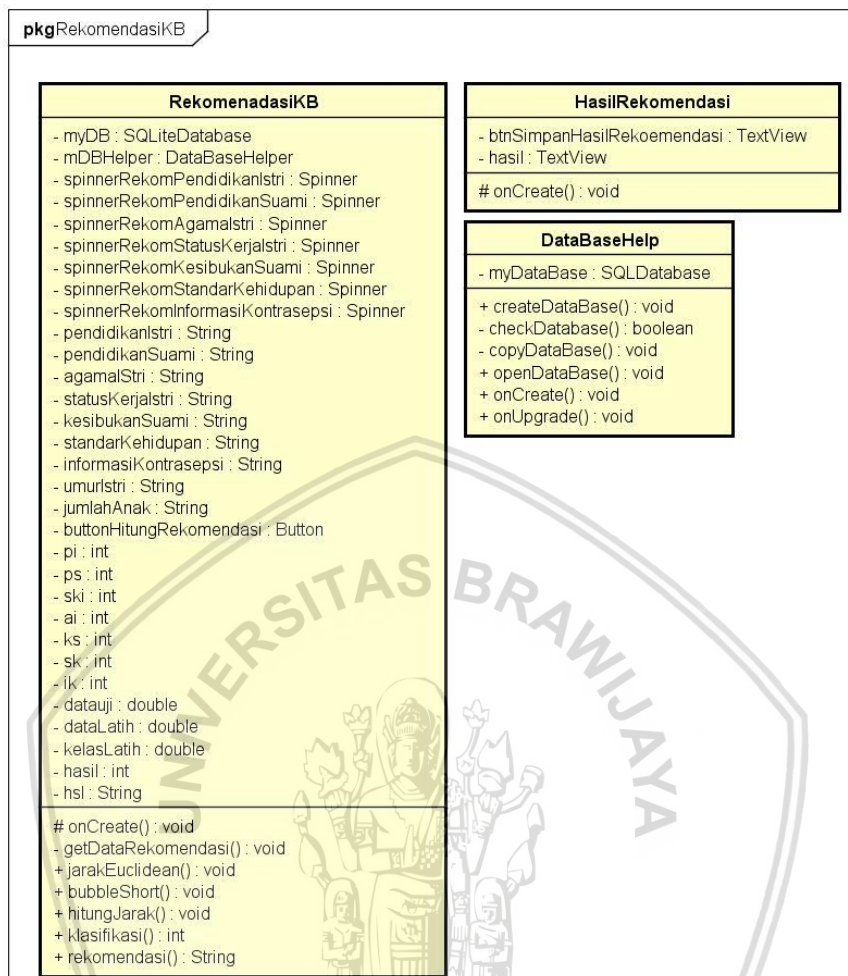
5.2.2.1 Perancangan Class Diagram Paket KalenderKB Iterasi 1



Gambar 5.34 Gambar Perancangan *Class Diagram* Paket Kalender KB Iterasi 1

Berdasarkan Gambar 5.9, karena fitur riwayat KB telah dihapus dan isi dari fitur riwayat KB digabung dengan fitur kalender KB, maka isi dari riwayat KB dipindah pada paket KalenderKB, yaitu sistol, diastole, dan beratbadan.

5.2.2.2 Perancangan Class Diagram Paket Rekomendasi KB



Gambar 5.35 Gambar Perancangan *Class Diagram* Paket Akseptor

Berdasarkan Gambar 5.10 paket RekomendasiKB memiliki tiga kelas, yaitu kelas RekomendasiKB sebagai *mainclass*, HasilRekomendasi yang digunakan untuk menampilkan hasil dari rekomendasi KB, dan DataBaseHelp sebagai penghubung antara *database* dan aplikasi.

5.2.3 Perancangan Antarmuka Iterasi 1

5.2.3.1 Hasil Evaluasi Iterasi 1

Berdasarkan perubahan pada analisis kebutuhan, perancangan basis data, dan perancangan *class diagram*, maka perancangan antarmuka pun berubah. Pada perancangan antarmuka iterasi 1 yang berubah adalah penambahan perancangan halaman rekomendasi KB dan penggabungan fitur riwayat KB dan Kalender KB. Berikut merupakan perancangan antarmuka halaman rekomendasi KB, hasil rekomendasi KB, halaman daftar riwayat kalender KB, dan buat kalender KB.

1. Halaman Rekomendasi KB

Keterangan:

1. Judul halaman Rekomendasi KB
2. Pertanyaan
3. Kolom untuk memilih jawaban
4. Tombol hitung

Gambar 5.36 Gambar Perancangan Antarmuka Halaman Rekomendasi KB

Pada Gambar 5.36 menjelaskan tentang perancangan antarmuka halaman rekomendasi KB. Pada halaman ini terdiri dari pertanyaan dan pilihan jawaban. Akseptor memilih jawaban yang sesuai dengan kondisinya saat ini dan memilih tombol hitung.

2. Halaman Hasil Rekomendasi KB

Keterangan:

1. Judul halaman hasil rekomendasi KB
2. Logo KBku
3. Hasil dari rekomendasi
4. Penjelasan rekomendasi KB

Gambar 5.37 Gambar Perancangan Antarmuka Halaman Hasil Rekomendasi KB

Pada Gambar 5.37 menjelaskan tentang perancangan antarmuka hasil dari rekomendasi KB. Pada halaman ini terdiri dari hasil rekomendasi KB dan penjelasan tentang hasil dari rekomendasi KB.

3. Halaman Daftar Riwayat Kalender KB

Pada Gambar 5.38 menjelaskan tentang perancangan antarmuka riwayat kalender yang telah dibuat sebelumnya. Pada halaman ini terdiri dari judul halaman dan daftar riwayat yang berisi tanggal, tekanan darah, berat badan pada pemberitahuan yang telah dilakukan sebelumnya.

Keterangan:

1. Judul dari halaman daftar riwayat kalender KB
2. Halaman daftar riwayat kalender KB
3. Kolom tekanan darah sistol
4. Kolom tekanan darah diastole
5. Tombol buat pemberitahuan KB

Gambar 5.38 Gambar Perancangan Antarmuka Halaman Riwayat Kalender KB

4. Halaman Buat Kalender KB

Keterangan:

1. Judul halaman buat kalender KB
2. Pilihan metode
3. Pilihan tanggal KB
4. Kolom untuk memasukkan tekanan darah sistol
5. Kolom untuk memasukkan tekanan darah diastole
6. Kolom untuk memasukkan berat badan
7. Tombol buat kalender KB

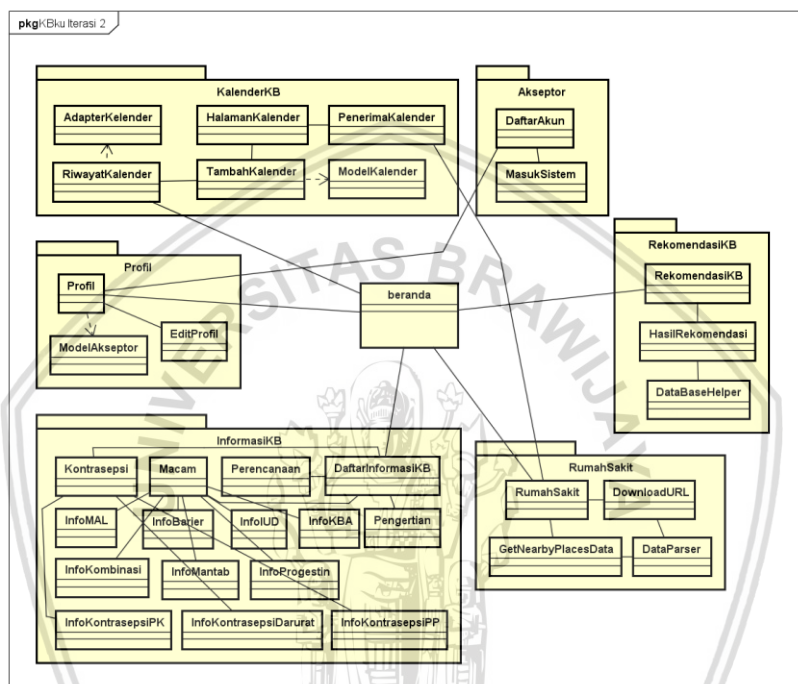
Gambar 5.39 Gambar Perancangan Antarmuka Halaman Buat Kalender KB

Pada Gambar 5.39 merupakan perancangan antarmuka halaman buat kalender KB. Pada halaman ini terdiri dari pilihan metode KB, pilihan tanggal terakhir untuk melakukan KB, kolom untuk memasukkan tekanan darah sebelumnya dan berat badan sebelumnya. Kemudian tombol simpan untuk menyimpan data kalender KB.

5.3 Perancangan Sistem Iterasi 2

Setelah dilakukan perancangan iterasi pertama maka didapatkan hasil bahwa pengguna KB menginginkan perubahan antarmuka pada halaman isi dari pemberitahuan KB, pengguna KB meminta ditambahkan tombol yang dapat berpindah ke halaman untuk mencari rumah sakit terdekat untuk memudahkan proses program KB. Berdasarkan perubahan tersebut maka terdapat perubahan pada perancangan *class diagram* dan perancangan antarmuka.

5.3.1 Perancangan Class Diagram Iterasi 2



Gambar 5.40 Gambar Perancangan *Class Diagram* Aplikasi KBku Iterasi 2

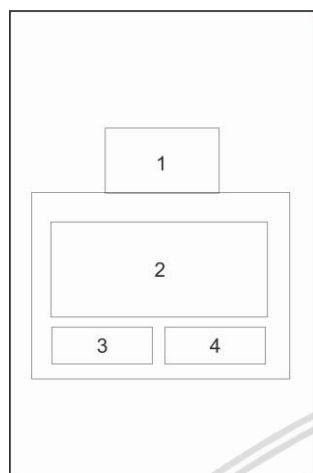
Pada Gambar 5.40 terjadi perubahan sedikit yaitu dihubungkannya paket kalenderKB dan paket RumahSakit, hal ini dilakukan karena pengguna KB meminta tombol yang menghubungkan halaman isi pemberitahuan KB dan halaman rumah sakit.

5.3.2 Perancangan Antarmuka Iterasi 2

5.3.2.1 Hasil Evaluasi Iterasi 2

Berdasarkan perubahan yang diinginkan pengguna KB pada halaman isi dari pemberitahuan untuk ditambahkannya tombol yang merujuk pada halaman mencari rumah sakit, maka berikut adalah perancangan antarmuka isi dari pemberitahuan KB.

1. Halaman Isi Pemberitahuan KB



Keterangan:

1. Logo KBku
2. Isi dari pemberitahuan KB
3. Tombol untuk mencari Rumah Sakit
4. Tombol Ok

Gambar 5.41 Gambar Perancangan Antarmuka Halaman Isi Pemberitahuan KB Iterasi 2

Pada Gambar 5.41 menjelaskan tentang perancangan antarmuka halaman isi dari pemberitahuan KB. Pada halaman ini terdapat penambahan tombol yang digunakan untuk membuka halaman rumah sakit untuk mencari rumah sakit terdekat.

5.3.3 Perancangan Algoritme

Sistem aplikasi perangkat bergerak KBku ini memiliki proses yang digunakan untuk menyelesaikan permasalahan seputar program KB, proses yang digunakan menggunakan implementasi algoritme. Algoritme yang digunakan untuk menyelesaikan mengambil data dari firebase, dan menampilkan hasil rekomendasi KB. Algoritme tersebut diimplementasikan kedalam kode program Java. Pada perancangan algoritme rekomendasi KB, peneliti menggunakan algoritme KNN yang sudah diteliti sebelumnya oleh saudara Christian yang mendapatkan kesimpulan bahwa tingkat akurasi yang diperoleh saat menggunakan algoritme KNN adalah sebesar 95% untuk rekomendasi KB. Untuk penjelasan secara detail tentang perancangan algoritme pada masing-masing permasalahan seputar program KB, dijelaskan pada subbab berikut.

5.3.3.1 Perancangan Algoritme Simpan Kalender KB

Perancangan algoritme simpan Kalender KB menjelaskan tentang proses penyimpanan kalender KB yang nantinya akan dijadikan sebuah kode program. Berikut merupakan penjelasan algoritme menyimpan kalender KB pada Gambar 5.42.

Algoritme Simpan Kalender KB

Deklarasi :

- Integer → tanggal, sistol, diastol, bb, metode

Deskripsi :

- Masukan : tanggal, sistol, diastole, bb, metode
- Proses :
 1. Inisialisasi method simpan
 2. Inisialisasi variable tanggal, sistol, diastol, bb, dan metode
 3. Jika sudah mengisi masukkan tanggal, sistol, diastol, bb, dan metode maka:
 - a. Menginisialisasi class ModelKalender untuk menyimpan data masukkan
 - b. Menyimpan data masukkan kedalam database PemberitahuanKB
 - c. Menampilkan pesan "Pemberitahuan telah ditambahkan"
 - d. Pindah halaman dari halaman tambahKalender ke RiwayatKalender
 - e. Memanggil method buatNotifikasi
 4. Jika ada data yang tidak terisi maka:
 - a. Menampilkan pesan "Masukkan data terlebih dahulu"
- Keluaran : Menyimpan data untuk melakukan waktu KB berikutnya

Gambar 5.42 Gambar Perancangan Algoritme Simpan Kalender KB

5.3.3.2 Perancangan Algoritme Buat Notifikasi

Perancangan algoritme buat notifikasi digunakan untuk membuat notifikasi waktu untuk melakukan program KB. Berikut merupakan penjelasan algoritme buat notifikasi pada Gambar 5.43.

Algoritme Buat Notifikasi

Deklarasi :

- String → Suntik 1 Bulan, Suntik 3 Bulan

Deskripsi :

- Masukan : Suntik 1 Bulan, Suntik 3 Bulan
- Proses :

1. Inisialisasi isi dari variable metode yang bernilai string
 2. Jika memilih suntik 1 bulan, maka:
 - a. Melakukan set alarmManager bahwa satu bulan kemudian dari tanggal awal dimasukkan, akan muncul pemberitahuan di halaman PenerimaKalender
 3. Jika memilih suntik 3 bulan, maka:
 - a. Melakukan set alarmManager bahwa tiga bulan kemudian dari tanggal awal dimasukkan, akan muncul pemberitahuan di halaman PenerimaKalender
- Keluaran : menampilkan tanggal pemberitahuan sesuai dengan apa yang telah dibuat

Gambar 5.43 Gambar Perancangan Algoritme Buat Notifikasi

5.3.3.3 Perancangan Algoritme Menampilkan Hasil Rekomendasi KB

Perancangan algoritme menampilkan hasil rekomendasi KB digunakan untuk menampilkan hasil dari perhitungan data yang telah dilakukan menggunakan algoritme k-NN. Berikut merupakan penjelasan algoritme menampilkan hasil rekomendasi KB pada Gambar 5.44.

Algoritme Menampilkan Hasil Rekomendasi KB

Deklarasi :

Int → k, out
 Double → array jarak
 String → hsl

Deskripsi :

- Masukan : -
- Proses :
 1. Inisialisasi nilai k = 3
 2. Inisialisasi variable out
 3. Inisialisasi array jarak
 4. Memanggil method hitungJarak dan mengurutkan menggunakan bubbleSort
 5. Hasil dari hitung jarak akan dimasukkan kedalam variable jarak

6. Memanggil method klasifikasi yang digunakan untuk menghitung hasil dari jarak dan k, kemudian disimpan dalam variable out
 7. Inisialisasi variable hsl
 8. Jika keluaran out bernilai 1
 - a. Maka hsl bernilai tidak memakai
 9. Jika keluaran out bernilai 2
 - a. Maka hsl bernilai jangka pendek
 10. Jika keluaran out bernilai 3
 - a. Maka hsl bernilai jangka panjang
 11. Mendapatkan nilai kembalian dari hsl
- Keluaran : menampilkan hasil dari rekomendasi KB

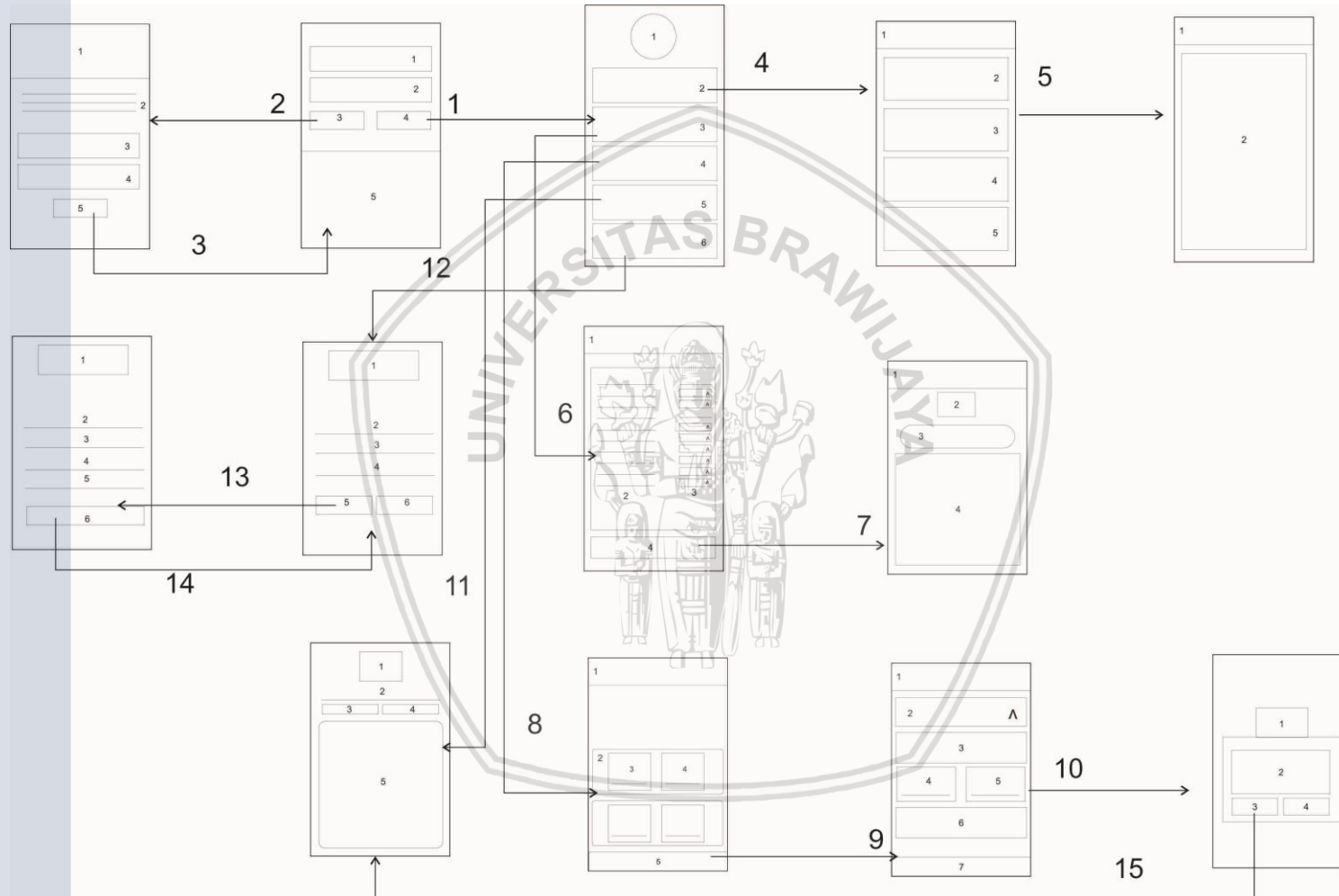
Gambar 5.44 Gambar Perancangan Algoritme Mencari Rumah sakit

5.3.4 Perancangan Antarmuka *Screenflow*

Pada Gambar 5.45 dijelaskan mengenai alur dari perancangan antarmuka pada aplikasi KBku. Perancangan antarmuka ini adalah perancangan antarmuka terakhir setelah dilakukan evaluasi yang mencapai 2 iterasi. Berikut merupakan penjelasan Gambar 5.45.

1. Sebelum pengguna KB menggunakan fitur yang terdapat pada aplikasi KBku, maka pengguna KB diharuskan untuk mengisi alamat *email* dan katasandi.
2. Jika pengguna KB belum memiliki akun maka diharuskan untuk mendaftarkan akun terlebih dahulu
3. Setelah memiliki akun, pengguna KB akan kembali kehalaman masuk, dan mengisi alamat *email* dan katasandi sesuai dengan yang sudah didaftarkan sebelumnya.
4. Ketika pengguna KB telah masuk ke dalam sistem maka, hak akses pengguna KB berubah menjadi akseptor, disini akseptor dapat melihat halaman beranda. Pada halaman beranda berisikan menu-menu yaitu menu Informasi KB, menu Rekomendasi KB, menu Kalender KB, menu Rumah Sakit, dan menu Profil. Pada tiap menu ini merujuk pada halaman selanjutnya. Ketika aksptor memilih menu informasi KB, maka akan muncul tampilan menu lagi, menu ini berisikan tentang macam-macam informasi KB.
5. Ketika akseptor memilih salah satu menu pada daftar informasi KB maka sistem akan menampilkan isi dari halaman informasi KB tersebut.

6. Ketika akseptor memilih menu rekomendasi KB, maka sistem akan menampilkan halaman rekomendasi yang berisikan *form* yang digunakan untuk mencari hasil dari rekomendasi KB
7. Ketika akseptor memilih tombol pada halaman rekomendasi KB maka sistem akan menampilkan halaman hasil dari rekomendasi KB
8. Akseptor memilih menu Kalender KB. Ketika akseptor memilih menu kalender KB, maka sistem akan menampilkan halaman riwayat kalender KB. Pada halaman ini berisikan tentang riwayat kalender KB yang telah akseptor lakukan sebelumnya. Pada halaman ini berisi tanggal KB sebelumnya, KB apa yang dilakukan sebelumnya, serta tekanan darah dan berat badan yang dimiliki pada saat melakukan KB sebelumnya.
9. Pada halaman riwayat kalender KB terdapat tombol yang digunakan untuk membuat kalender KB baru. Ketika akseptor memilih tombol ini maka sistem akan menampilkan halaman buat kalender KB baru. Pada halaman ini berisi *form* yang digunakan untuk mengisi data pada saat melakukan pembuatan kalender KB baru. Pada *form* ini berisi metode, disini akseptor dapat memilih metode apa yang digunakan untuk mendapatkan pemberitahuan KB selanjutnya. Kemudian akseptor memilih tanggal untuk melakukan KB. Kemudian akseptor diminta untuk memasukkan tekanan darah dan berat badan terakhir. Kemudian akseptor dapat menyimpan data tersebut pada tombol simpan yang telah tersedia pada halaman buat kalender KB.
10. Ketika sudah waktunya untuk melakukan program KB selanjutnya, maka sistem akan menampilkan pemberitahuan untuk melakukan program KB.
11. Akseptor memilih menu rumah sakit. Pada menu ini akseptor dapat melihat rumah sakit terdekat dan dapat mencari rute untuk menuju rumah sakit yang paling terdekat dari posisinya saat itu.
12. Akseptor memilih menu profil. Pada saat akseptor memilih menu profil, maka sistem akan menampilkan halaman profil, pada halaman ini terdapat informasi yang akan dimasukkan oleh akseptor pada saat akseptor memilih tombol ubah profil. Didalam halaman ini juga terdapat tombol keluar dari sistem jika akseptor ingin keluar dari aplikasi.
13. Ketika akseptor memilih tombol ubah profil maka sistem akan menampilkan halaman ubah profil. Pada halaman ini akseptor dapat melakukan ubah profil dengan memasukkan nama, alamat dan nomor *handphone*. Untuk melakukan penyimpanan, maka akseptor memilih tombol simpan.
14. Ketika akseptor memilih tombol simpan maka Sistem akan menampilkan halaman profil kembali.
15. Ketika akseptor menerima pemberitahuan KB maka akseptor dapat memilih tombol untuk mencari rumah sakit terdekat. Ketika akseptor memilih tombol ini maka sistem akan langsung menghubungkan dengan halaman rumah sakit.



Gambar 5.45 Gambar Perancangan Antarmuka *Screenflow*

5.4 Implementasi Sistem

Pada subbab implementasi sistem menjelaskan tentang proses eksekusi perancangan yang telah dibuat. Pada subbab implementasi sistem terdiri dari spesifikasi sistem, batasan implementasi, implementasi basis data, implementasi *class*, implementasi kode program, dan implementasi antarmuka.

5.4.1 Spesifikasi Sistem

Spesifikasi sistem membahas tentang hasil dari analisa kebutuhan, perancangan dan evaluasi yang nantinya diimplementasikan pada perangkat keras dan juga membutuhkan perangkat lunak. Perangkat keras dan perangkat lunak tersebut dispesifikasikan untuk mengetahui standar yang diperlukan pada saat membangun aplikasi.

5.4.1.1 Spesifikasi Perangkat Keras

Untuk melakukan proses implementasi aplikasi KBku dibutuhkan perangkat keras seperti *laptop* dan *smartphone*. *Laptop* sendiri digunakan untuk melakukan proses pembuatan kode program yang nantinya akan di tampilkan di *smartphone*. Spesifikasi perangkat keras *laptop* yang digunakan untuk melakukan implementasi dapat dilihat pada Tabel 5.6.

Tabel 5.6 Spesifikasi Perangkat Keras

Keterangan	Spesifikasi
System Model	ASUS K401L
Processor	Intel® Core™ i5-5200U CPU @ 2.20GHz 2.19GHz
Memory	4GB RAM, 1TB HDD
Display	Nvidia GEFORCE 940m

5.4.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak pada *laptop* yang digunakan untuk melakukan implementasi dapat dilihat pada Tabel 5.7.

Tabel 5.7 Spesifikasi Perangkat Lunak

ASUS K401L	
<i>Operating System</i>	Mircrosoft Windows 10 Pro 64bit
<i>Programming Language</i>	Java
<i>Programming Tool</i>	Android Studio 3.1
<i>Android SDK Tools</i>	26.0.1
<i>Gradle Version</i>	4.1
<i>Database Management System</i>	Firebase realtime database SDK 11.6.2

5.4.1.3 Spesifikasi Perangkat Bergerak

Spesifikasi perangkat bergerak yang digunakan untuk melakukan implementasi dapat dilihat pada Tabel 5.8.

Tabel 5.8 Spesifikasi Perangkat Bergerak

Keterangan	Spesifikasi
System Model	Samsung S7 edge duos – G9350
Processor	Snapdragon 820
Memory	32 GB, 4GB RAM
Display	2560 x 1440 pixels, 5.5" Super AMOLED Screen

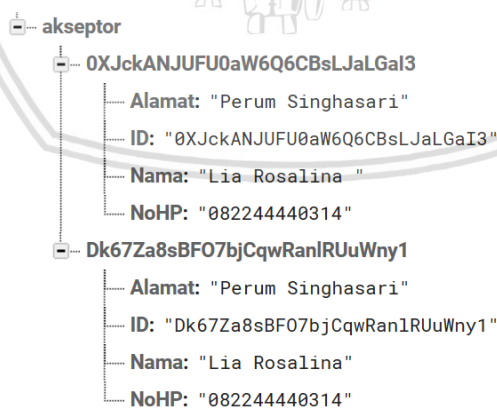
5.4.2 Batasan Implementasi

Dalam melakukan implementasi aplikasi KBku terdapat beberapa batasan implementasi, batasan implementasi tersebut terdiri dari:

1. Aplikasi dijalankan pada *smartphone* dengan sistem operasi Android dengan API 19 keatas.
2. Aplikasi dibangun dengan menggunakan Bahasa pemrograman java.
3. Aplikasi KBku dijalankan pada *smartphone* yang terkoneksi dengan *internet*.

5.4.3 Implementasi Basis Data

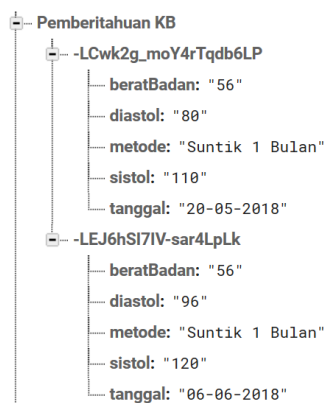
5.4.3.1 Basis Data Akseptor



Gambar 5.46 Implementasi Basis Data Akseptor

Gambar 5.46 merupakan implementasi dari basis data akseptor digunakan untuk menyimpan isi data Alamat, ID, Nama, NoHP. Atribut yang dimiliki oleh basis data akseptor adalah Alamat, ID, Nama, NoHP.

5.4.3.2 Basis Data Pemberitahuan KB



Gambar 5.47 Implementasi Basis Pemberitahuan KB

Gambar 5.47 merupakan implementasi dari basis data kalender KB yang digunakan untuk menyimpan isi dari beratbadan, diastole, metode, sistol dan tanggal KB. Atribut yang dimiliki oleh basis data Pemberitahuan KB adalah beratbadan, diastole, metode, sistol dan tanggal KB.

5.4.3.3 Basis Data Rekomendasi KB

▼ RekomendasiKB		CREATE TABLE "RekomendasiKB" ("id" INTEGER, 'v1' INTEGER, 'v2' INTEGER, 'v3' INTEGER, 'v4' INTEGER, 'v5' INTEGER, 'v6' INTEGER, 'v7' INTEGER, 'v8' INTEGER, 'v9' INTEGER, kelas INTEGER)
id	INTEGER	'id' INTEGER
v1	INTEGER	'v1' INTEGER
v2	INTEGER	'v2' INTEGER
v3	INTEGER	'v3' INTEGER
v4	INTEGER	'v4' INTEGER
v5	INTEGER	'v5' INTEGER
v6	INTEGER	'v6' INTEGER
v7	INTEGER	'v7' INTEGER
v8	INTEGER	'v8' INTEGER
v9	INTEGER	'v9' INTEGER
kelas	INTEGER	'kelas' INTEGER

Gambar 5.48 Implementasi Struktur Basis Data Rekomendasi KB

Gambar 5.48 merupakan implementasi dari basis data Rekomendasi KB yang digunakan untuk menyimpan data latih dari rekomendasi KB. Atribut yang dimiliki oleh basis data Rekomendasi KB adalah `_id`, `v1`, `v2`, `v3`, `v4`, `v5`, `v6`, `v7`, `v8`, `v9`, dan `kelas`. Semua atribut yang berada pada basis data rekomendasiKB bertipe INTEGER. Gambar 5.49 merupakan isi data dari masing-masing atribut.

Table: RekomendasiKB

New Record Delete Record

	_id	v1	v2	v3	v4	v5	v6	v7
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	24	2	3	3	1	1	2
2	2	45	1	3	10	1	1	3
3	3	43	2	3	7	1	1	3
4	4	42	3	2	9	1	1	3
5	5	36	3	3	8	1	1	3
6	6	19	4	4	0	1	1	3
7	7	38	2	3	6	1	1	3
8	8	21	3	3	1	1	0	3
9	9	27	2	3	3	1	1	3
10	10	45	1	1	8	1	1	2
11	11	38	1	3	2	1	0	3
12	12	42	1	4	4	1	1	1
13	13	44	4	4	1	1	0	1
14	14	42	2	4	1	1	0	3
15	15	38	3	4	2	1	1	2

Gambar 5.49 Implementasi Basis Data Rekomendasi KB

5.4.4 Implementasi Class

Dari setiap class yang dirancang pada tahap perancangan direalisasikan pada beberapa *file* program dengan ekstensi *.java. Daftar implementasi *class* yang telah dibuat ditunjukkan pada Tabel 5.9.

Tabel 5.9 Implementasi Class

No	Package	Nama Class / Interface	Nama File Program
1	Akseptor	DaftarAkun	DaftarAkun.java
2	Akseptor	MasukSistem	MasukSistem.java
3	InformasiKB	DaftarInformasiKB	DaftarInformasiKB.java
4	InformasiKB	InfoBarrier	InfoBarrier.java
5	InformasiKB	InfoIUD	InfoIUD.java
6	InformasiKB	InfoKBA	InfoKBA.java
7	InformasiKB	InfoKombinasi	InfoKombinasi.java
8	InformasiKB	InfoKontrasepsiDarurat	InfoKontrasepsiDarurat.java
9	InformasiKB	InfoKontrasepsiPK	InfoKontrasepsiPK.java
10	InformasiKB	InfoKontrasepsiPP	InfoKontrasepsiPP.java
11	InformasiKB	InfoMAL	InfoMAL.java
12	InformasiKB	InfoMantab	InfoMantab.java
13	InformasiKB	InfoProgestin	InfoProgestin.java
14	InformasiKB	Kontrasepsi	Kontrasepsi.java
15	InformasiKB	Macam	Macam.java

16	InformasiKB	Pengertian	Pengertian.java
17	InformasiKB	Pernecanaan	Pernecanaan.java
18	KalenderKB	AdapterKalender	AdapterKalender.java
19	KalenderKB	HalamanKalender	HalamanKalender.java
20	KalenderKB	ModelKalender	ModelKalender.java
21	KalenderKB	PenerimaKalender	PenerimaKalender.java
22	KalenderKB	RiwayatKalender	RiwayatKalender.java
23	KalenderKB	TambahKalender	TambahKalender.java
24	KalenderKB	ModelKalender	ModelKalender.java
25	Profil	Editprofil	Editprofil.java
26	Profil	ModelAkseptor	ModelAkseptor.java
27	Profil	Profil	Profil.java
28	RekomendasiKB	DataBaseHelper	DataBaseHelper.java
29	RekomendasiKB	HasilRekomendasi	HasilRekomendasi.java
30	RekomendasiKB	RekomendasiKB	RekomendasiKB.java
31	RumahSakit	DataParser	DataParser.java
32	RumahSakit	DownloadURL	DownloadURL.java
33	RumahSakit	GetNearbyPlacesData	GetNearbyPlacesData.java
34	RumahSakit	RumahSakit	RumahSakit.java
35		beranda	beranda.java

5.4.5 Implementasi Kode Program

5.4.5.1 Implementasi *Method* simpan()

Method simpan digunakan oleh untuk menyimpan data ketika membuat kalender KB baru. Kode program *method* simpan ditunjukkan oleh Gambar 5.50.

```

1  private void simpan() {
2      String tanggal = Tanggal.getText().toString();
3      String sistol = Sistolik.getText().toString();
4      String diastol = Diastolik.getText().toString();
5      String bb = BeratBadan.getText().toString();
6      String metode = Metode.getSelectedItem().toString();
7
8      if (!TextUtils.isEmpty(bb)) {
9          String id = dbPemberitahuanKB.push().getKey();
10         ModelKalender tahu = new ModelKalender(metode, tanggal,
11         sistol, diastol, bb);
12         dbPemberitahuanKB.child(id).setValue(tahu);
13         Sistolik.setText("");

```

14	Diastolik.setText("");
15	BeratBadan.setText("");
16	Toast.makeText(this, "Pemberitahuan telah ditambahkan",
17	Toast.LENGTH_LONG).show();
18	Intent i = new Intent (TambahKalender.this,
19	RiwayatKalender.class);
20	startActivity(i);
21	buatNotifikasi();
22	} else {
23	Toast.makeText(this, "Masukkan data terlebih dahulu",
24	Toast.LENGTH_LONG).show();
25	}
26	}

Gambar 5.50 Implementasi *Method* simpan()

Penjelasan kode program *method* simpan ditunjukkan pada Tabel 5.10.

Tabel 5.10 Penjelasan *Method* simpan()

Nomor Kode Program	Penjelasan
1	Inisialisasi metode simpan
2-6	Deklarasi objek untuk melakukan pengambilan data yang telah dimasukkan oleh akseptor dan merubahnya menjadi String
8-21	Kondisi jika <i>vairiable</i> bb tidak kosong, akan melakukan inisialisasi <i>variable</i> id bertipe String untuk membuat id pada <i>database</i> . Kemudian melakukan inisialisasi objek tahu yang merujuk pada ModelKalender. Mengirimkan value objek tahu pada <i>variable</i> dbPemberitahuanKB, mengubah beberapa text pada <i>variable</i> menjadi kosong dan menampilkan teks jika proses berhasil.
22-26	Jika data tidak diisi dengan lengkap, maka akan menampilkan teks pemberitahuan.

5.4.5.2 Implementasi *Method* buatNotifikasi()

Method buatNotifikasi digunakan oleh untuk membuat pemberitahuan KB dengan menggunakan alarmManager. Kode program *method* buatNotifikasi ditunjukkan oleh Gambar 5.51.

1	private void buatNotifikasi(){
2	String metode = Metode.getSelectedItem().toString();
3	
4	if (metode.equals("Suntik 1 Bulan")) {
5	AlarmManager alarmManager = (AlarmManager)
6	getSystemService(ALARM_SERVICE);
7	Intent notificationIntent = new Intent(this,
8	PenerimaKalender.class);
9	PendingIntent broadcast =
10	PendingIntent.getBroadcast(this, 100, notificationIntent,
11	PendingIntent.FLAG_UPDATE_CURRENT);

12	Calendar cal = Calendar.getInstance();
13	cal.add(Calendar.MONTH, 1);
14	alarmManager.setExact(AlarmManager.RTC_WAKEUP,
15	cal.getTimeInMillis(), broadcast);
16	} else (metode.equals("Suntik 3 Bulan")) {
17	AlarmManager alarmManager = (AlarmManager)
18	getSystemService(Context.ALARM_SERVICE);
19	Intent notificationIntent = new Intent(this,
20	PenerimaKalender.class);
21	PendingIntent broadcast =
22	PendingIntent.getBroadcast(this, 100, notificationIntent,
23	PendingIntent.FLAG_UPDATE_CURRENT);
24	Calendar cal = Calendar.getInstance();
25	cal.add(Calendar.MONTH, 3);
26	alarmManager.setExact(AlarmManager.RTC_WAKEUP,
27	cal.getTimeInMillis(), broadcast);
28	}
29	}

Gambar 5.51 Implementasi Method buatNotifikasi()

Penjelasan kode program *method* buatNotifikasi ditunjukkan pada Tabel 5.11.

Tabel 5.11 Penjelasan Method buatNotifikasi()

Nomor Kode Program	Penjelasan
1	Inisialisasi metode buatNotifikasi
2-3	Deklarasi objek untuk melakukan pengambilan data masukan dari pengguna dan merubahnya menjadi string.
4-15	Jika pengguna KB memilih suntuk 1 bulan maka nilai dari masukkan tersebut akan di set pada AlarmManager untuk menampilkan pemberitahuan suntuk 1 bulan
16-29	Jika pengguna KB memilih suntuk 3 bulan maka nilai dari masukkan tersebut akan di set pada AlarmManager untuk menampilkan pemberitahuan suntuk 3 bulan

5.4.5.3 Implementasi Method Rekomendasi()

Method rekomendasi digunakan oleh untuk menghitung hasil dari rekomendasi. Kode program *method* rekomendasi ditunjukkan oleh Gambar 5.52.

1	public String rekomendasi() {
2	int k = 1;
3	int out;
4	//inisialisasi
5	double jarak[][] = new double[1473][3];
6	//hitung jarak
7	jarak = hitungJarak(0);
8	bubbleSort(jarak);
9	jarak = Arrays.copyOf(jarak, k);
10	out = klasifikasi(jarak, k);

11	
12	String hsl = null;
13	if (out == 1) {
14	hsl = "tidak memakai";
15	} else if (out == 2) {
16	hsl = "jangka pendek";
17	} else if (out == 3) {
18	hsl = "jangka panjang";
19	} else {
20	hsl = "error";
21	}
22	return hsl;
23	}

Gambar 5.52 Implementasi Method rekomendasi()

Penjelasan kode program *method* rekomendasi ditunjukkan pada Tabel 5.12.

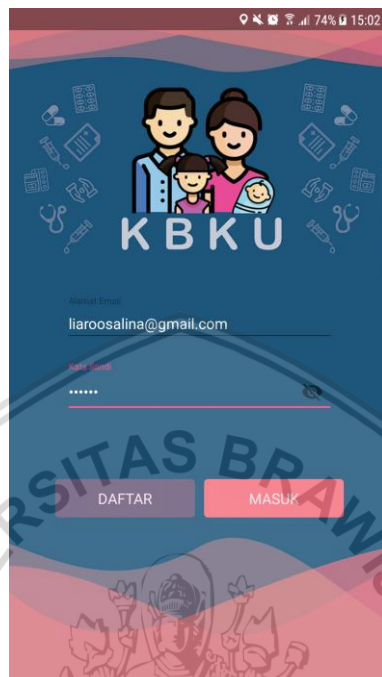
Tabel 5.12 Penjelasan Method buatNotifikasi()

Nomor Kode Program	Penjelasan
1	Inisialisasi metode rekomendasi
2	inisialisasi variabel k sebagai tetangga terdekat, nilainya bebas minimal 1 dan maksimal sesuai jumlah data latih (1473).
3	inisialisasi variabel out untuk menyimpan hasil perhitungan klasifikasi knn seperti di baris ke 10
5	inisialisasi jarak berupa array 2d untuk menyimpan jarak setiap data latih.
7	Nilai jarak didapat dari hasil perhitungan yang ada di method hitungjarak(0).
8	Memanggil method Bubblesort(), parameternya berupa array jarak.
9	Copy data array jarak sesuai dengan jumlah k ditentukan diawal.
12	Inisialisasi hsl, karena variable hsl berupa string dan masih kosong sedangkan syarat agar bisa digunakan dalam suatu operasi harus ada nilainya makanya diberi nilai null.
13-21	Ini untuk menentukan hasil akhir klasifikasinya. Mengubah dari angka (int) ke kalimat (string).
22	Var hsl di return sebagai nilai akhir yang dihasilkan.

5.4.6 Implementasi Antarmuka

5.4.6.1 Detail Implementasi Antarmuka

(a) Antarmuka Halaman Masuk ke Sistem

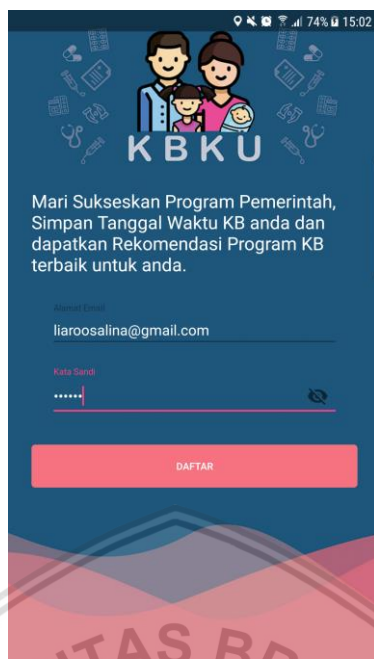


Gambar 5.53 Implementasi Masuk Ke Sistem

Pada Gambar 5.53 merupakan implementasi antarmuka masuk kedalam sistem. Agar pengguna KB dapat masuk kedalam sistem, maka pengguna KB diharuskan mengisi form masuk ke sistem yang berisi alamat *email* serta kata sandi dan memilih tombol masuk. Namun jika pengguna KB belum memiliki akun, maka pengguna KB diharuskan memilih tombol daftar untuk mendaftarkan akun.

(b) Antarmuka Halaman Daftar Akun

Pada Gambar 5.54 merupakan implementasi antarmuka masuk kedalam sistem. Agar pengguna KB dapat masuk kedalam sistem, maka pengguna KB diharuskan mengisi form masuk ke sistem yang berisi alamat *email* serta kata sandi dan memilih tombol masuk. Namun jika pengguna KB belum memiliki akun, maka pengguna KB diharuskan memilih tombol daftar untuk mendaftarkan akun.



Gambar 5.54 Implementasi Daftar Akun

(c) Antarmuka Halaman Beranda



Gambar 5.55 Implementasi Halaman Beranda

Pada Gambar 5.55 menjelaskan tentang implementasi antarmuka halaman beranda. Pada halaman beranda ini terdapat logo aplikasi KBku pada bagian atas

dan terdapat lima menu yaitu informasi KB, Rekomendasi KB, Kalender KB, Rumah sakit, serta Profil. Didalam menu tersebut terdapat *icon* dan judul menu

(d) Antarmuka Halaman Daftar Informasi KB



Gambar 5.56 Implementasi Daftar Informasi KB

Pada Gambar 5.56 merupakan implementasi halaman daftar informasi KB. Pada halaman daftar informasi KB ini terdiri dari beberapa informasi KB yang bisa dipilih oleh akseptor sesuai dengan berita yang diinginkan. Pada halaman informasi KB memiliki menu yang berisi Gambar serta judul dari berita.

(e) Antarmuka Halaman Detail Informasi KB

Pada Gambar 5.57 merupakan halaman detail informasi KB. Pada halaman detail informasi KB terdiri dari judul halaman informasi KB dan isi informasi KB.



Pengertian KB



Keluarga Berencana (KB) merupakan suatu program pemerintah yang dirancang untuk menyeimbangkan antara kebutuhan dan jumlah penduduk. Program keluarga berencana oleh pemerintah adalah agar keluarga sebagai unit terkecil kehidupan bangsa diharapkan menerima Norma Keluarga Kecil Bahagia dan Sejahtera (NKKBS) yang berorientasi pada pertumbuhan yang seimbang. Gerakan Keluarga Berencana Nasional Indonesia telah berumur sangat lama yaitu pada tahun 70-an dan masyarakat dunia menganggap berhasil menurunkan angka kelahiran yang bermakna. Perencanaan jumlah keluarga dengan pembatasan yang bisa dilakukan dengan penggunaan alat-alat kontrasepsi atau penangulangan kelahiran seperti kondom, spiral, IUD, dan sebagainya.

Tujuan Keluarga berencana (KB)

• Tujuan Umum

Meningkatkan kesejahteraan ibu dan anak dalam rangka mewujudkan NKKBS (Normal Keluarga Kecil Bahagia Sejahtera) yang menjadi dasar terwujudnya masyarakat yang

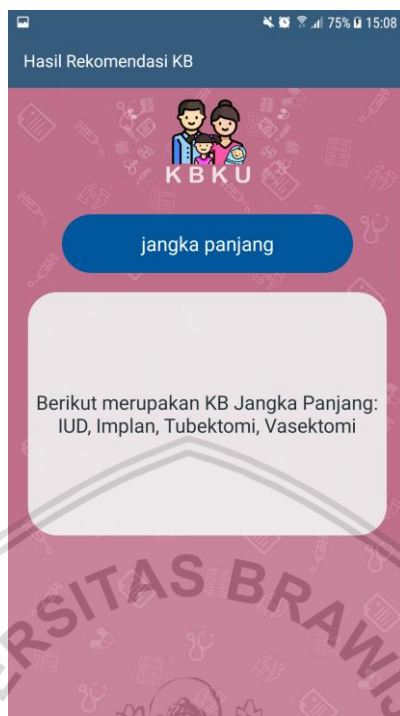
Gambar 5.57 Implementasi Informasi KB

(f) Antarmuka Halaman Rekomendasi KB

Gambar 5.58 Implementasi Rekomendasi KB

Pada Gambar 5.58 implementasi antarmuka halaman rekomendasi KB. Pada halaman ini terdiri dari pertanyaan dan pilihan jawaban. Akseptor memilih jawaban yang sesuai dengan kondisinya saat ini dan memilih tombol hitung.

(g) Antarmuka Halaman Hasil Rekomendasi KB



Gambar 5.59 Implementasi Hasil Rekomendasi KB

ada Gambar 5.59 menjelaskan tentang implementasi antarmuka hasil dari rekomendasi KB. Pada halaman ini terdiri dari hasil rekomendasi KB dan penjelasan tentang hasil dari rekomendasi KB.

(h) Antarmuka Halaman Kalender KB

Pada Gambar 5.60 merupakan implementasi antarmuka halaman buat kalender KB. Pada halaman ini terdiri dari pilihan metode KB, pilihan tanggal terakhir untuk melakukan KB, kolom untuk memasukkan tekanan darah sebelumnya dan berat badan sebelumnya. Kemudian tombol simpan untuk menyimpan data kalender KB.

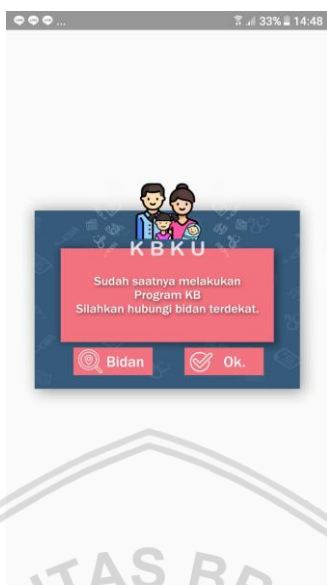
Gambar 5.60 Implementasi Kalender KB

(i) Antarmuka Halaman Daftar Riwayat Kalender KB

Gambar 5.61 Implementasi Daftar Riwayat Kalender KB

Pada Gambar 5.61 menjelaskan tentang implementasi antarmuka riwayat kalender yang telah dibuat sebelumnya. Pada halaman ini terdiri dari judul halaman dan daftar riwayat yang berisi tanggal, tekanan darah, berat badan pada pemberitahuan yang telah dilakukan sebelumnya.

(j) Antarmuka Halaman Pemberitahuan KB

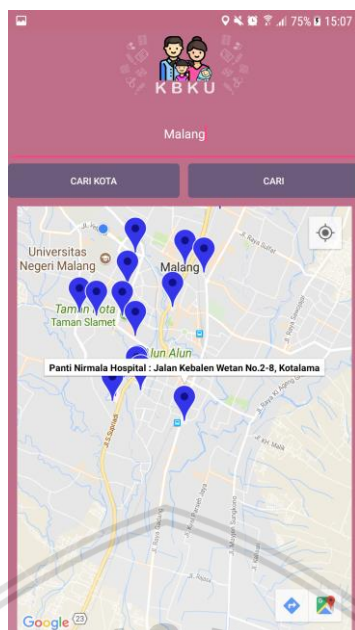


Gambar 5.62 Implementasi Pemberitahuan KB

Pada Gambar 5.62 menjelaskan tentang implementasi antarmuka halaman isi dari pemberitahuan KB. Pada halaman ini terdapat penambahan tombol yang digunakan untuk membuka halaman rumah sakit untuk mencari rumah sakit terdekat.

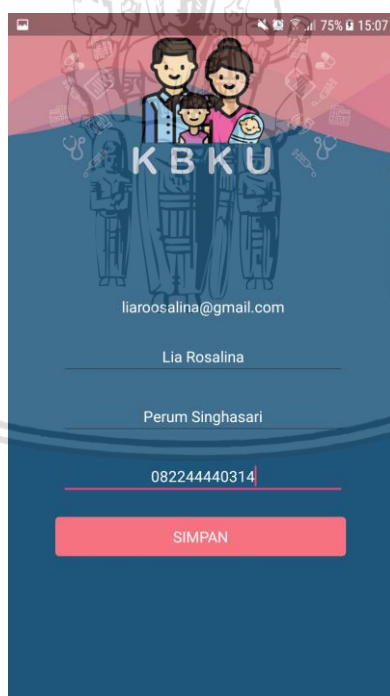
(k) Antarmuka Halaman Rumah Sakit Terdekat

Gambar 5.63 merupakan gambar implementasi antarmuka untuk mencari rumah sakit terdekat. Pada halaman ini terdiri dari logo KBku, kolom untuk memasukkan kota yang ingin dicari, kemudian tombol untuk mencari kota saat ini dan tombol mencari rumah sakit terdekat. Dan pada bagian bawah tombol terdapat peta yang akan menampilkan rumah sakit terdekat dan rute untuk menuju rumah sakit yang dituju.



Gambar 5.63 Implementasi Rumah Sakit

(I) Antarmuka Halaman Ubah Profil

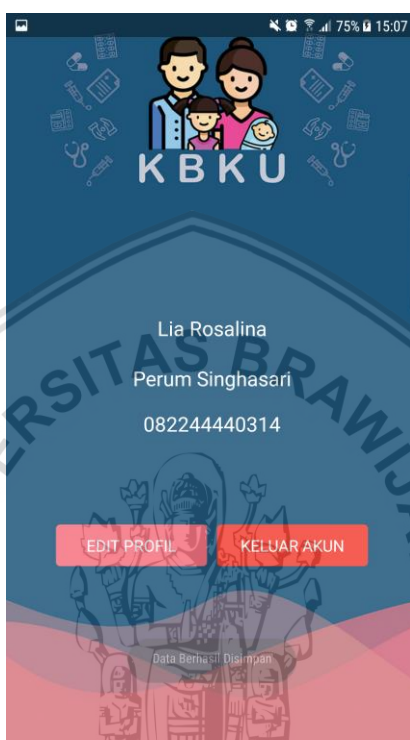


Gambar 5.64 Implementasi Halaman Ubah Profil

Gambar 2.64 merupakan implementasi halaman ubah profil. Pada bagian atas halaman uah profil terdapat logo aplikasi KBku. Kemudian pada bagian tenagh halaman terdapat form yang digunakan untuk mengisi data yang akan disimpan dan akan ditampilkan pada profil. Selanjutnya terdapat tombol simpan yang digunakan untuk menyimpan data yang telah diisikan.

(m) Antarmuka Halaman Profil

Pada Gambar 5.65 merupakan implementasi halaman profil, pada bagian atas halaman profil terdapat logo KBku. Kemudian terdapat isi dari profil pengguna. Pada bagian bawah halaman terdapat dua tombol yaitu tombol edit profil dan tombol keluar sistem.



Gambar 5.65 Implementasi Halaman Profil

BAB 6 PENGUJIAN

Pada bab pengujian ini akan dibahas tahapan mengenai pengujian dari pembuatan aplikasi KBku. Pengujian yang dilakukan meliputi pengujian fungsional dan pengujian non-fungsional. Pengujian fungsional merupakan pengujian yang digunakan untuk menguji kebutuhan fungsional pada sistem. Untuk pengujian non-fungsional sendiri merupakan pengujian yang digunakan untuk menguji kebutuhan non-fungsional pada sistem. Proses pengujian pada penelitian KBku ini terdiri dari pengujian unit, pengujian validasi, pengujian *usability*, dan pengujian *compatibility*.

6.1 Pengujian Fungsional

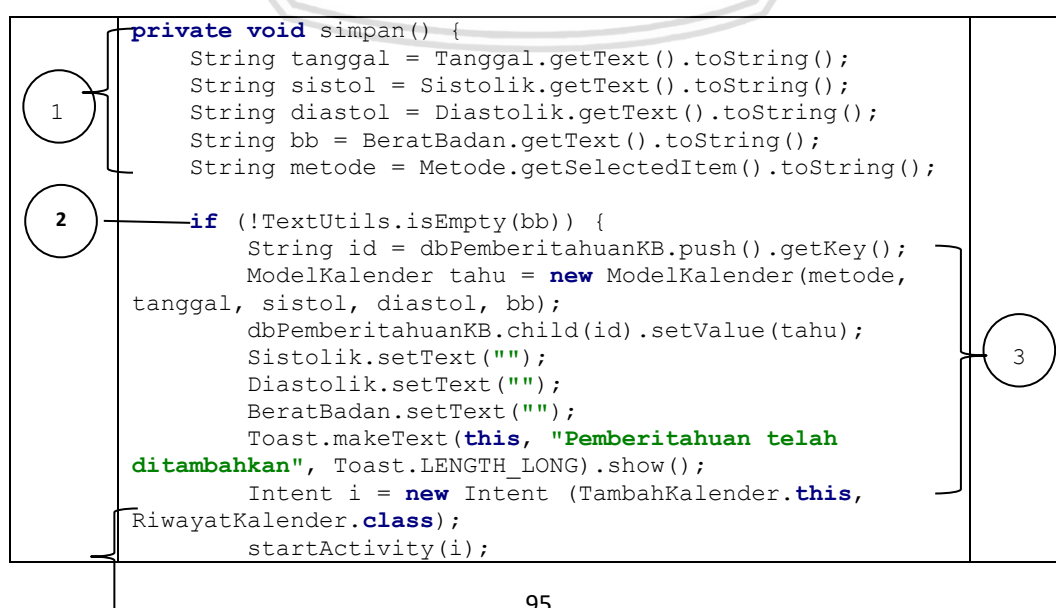
Pengujian fungsional digunakan untuk menguji seluruh kebutuhan fungsional yang telah dikembangkan dalam sistem. Pengujian ini terbagi menjadi pengujian unit, pengujian validasi. Pada tahap ini pengujian unit hanya dilakukan pada *method* dari fungsi utama sistem saja.

6.1.1 Pengujian Unit

Pengujian unit aplikasi KBku ini menggunakan *white-box testing* dengan teknik *basis path testing*. Teknik *basis path testing* ini memodelkan algoritme pada suatu *flow graph* dan menentukan jumlah kompleksitas siklomatis, menentukan sebuah basis set dari jalur independen dan memberi kasus uji pada setiap basis set yang telah ditentukan sebelumnya.

6.1.1.1 Pengujian *Method* Simpan Kalender KB

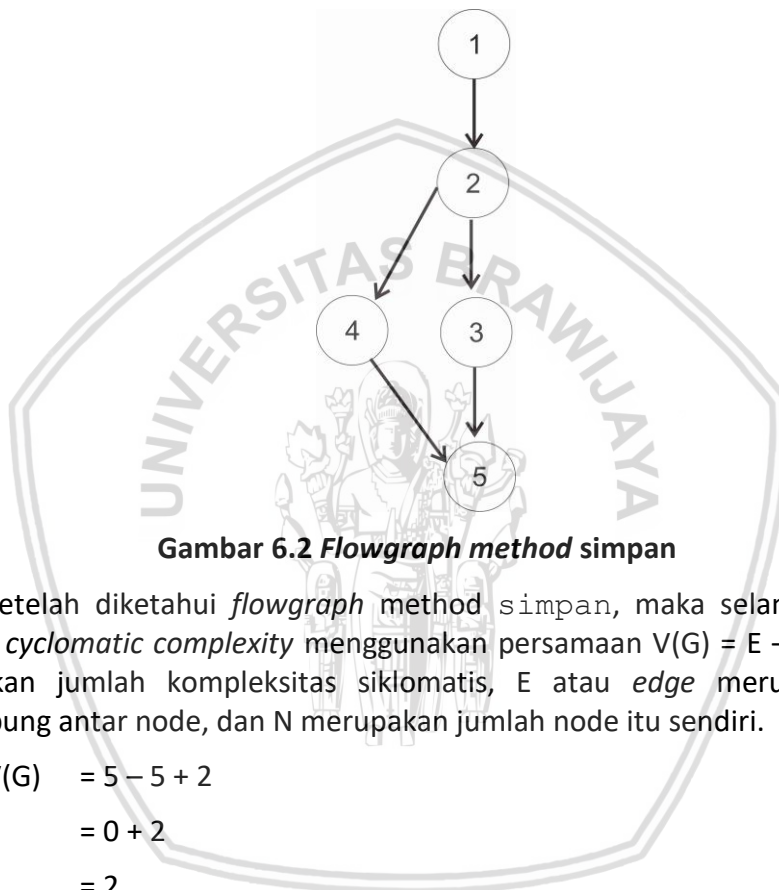
Pengujian unit *method* simpan kalenderKB digunakan untuk melakukan pengujian pada *method* `simpan()`. *Method* `simpan()` sendiri digunakan untuk melakukan penyimpanan kalender KB untuk melakukan waktu KB. Didalam *method* `simpan()` akan diinisialisasi masukkan yang telah dimasukkan oleh akseptor. Kode program *method* `simpan()` ditunjukkan oleh Gambar 6.1.



3	<pre> buatNotifikasi(); } else { Toast.makeText(this, "Masukkan data terlebih dahulu", Toast.LENGTH_LONG).show(); } </pre>	4
5	}	

Gambar 6.1 Kode Program Method *simpan*

Berdasarkan proses pengujian unit yang telah dilakukan sebelumnya, maka *method* *simpan()* akan menghasilkan *flowgraph* seperti yang ditunjukkan oleh Gambar 6.2.



Gambar 6.2 Flowgraph method *simpan*

Setelah diketahui *flowgraph* method *simpan*, maka selanjutnya akan dihitung *cyclomatic complexity* menggunakan persamaan $V(G) = E - N + 2$. $V(G)$ merupakan jumlah kompleksitas siklomatis, E atau *edge* merupakan garis penghubung antar node, dan N merupakan jumlah node itu sendiri.

$$\begin{aligned}
 V(G) &= 5 - 5 + 2 \\
 &= 0 + 2 \\
 &= 2
 \end{aligned}$$

Berdasarkan hasil dari perhitungan kompleksitas siklomatis, maka didapatkan dua basis set jalur *independent*, yaitu:

Jalur 1 : 1 – 2 – 3 – 5

Jalur 2 : 1 – 4 – 5

Setelah mendapatkan urutan node dari tiap jalur *independent* maka tahap selanjutnya adalah melakukan pengujian tiap jalurnya. Pengujian terhadap tiap jalur tersebut ditunjukkan oleh Tabel 6.1.

Tabel 6.1 Kasus Uji Pengujian Unit *Method* simpan

Jalur	Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan
1	Ketika sudah memasukkan data berupa tanggal, sistol, diastol, bb, dan metode	<i>Method</i> simpan() dapat melakukan proses penyimpanan kedalam database	<i>Method</i> simpan() dapat melakukan penyimpanan data pada firebase database
2	Ketika diminta memasukkan data, namun ada data belum dimasukkan atau masih ada data yang kosong	<i>Method</i> simpan() akan menampilkan pemberitahuan "masukkan data terlebih dahulu"	<i>Method</i> simpan() menampilkan pemberitahuan untuk memasukan data

6.1.1.2 Pengujian *Method* buatNotifikasi

Pengujian unit *Method* buatNotifikas() digunakan untuk membuat pemberitahuan KB. Kode program *method* buatNotifikasi() ditunjukkan oleh Gambar 6.3.

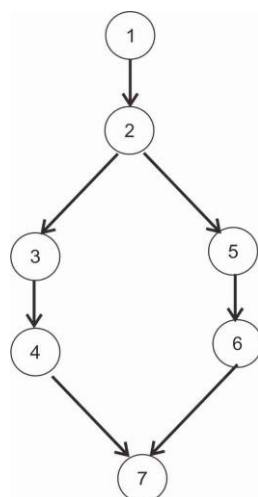
```

1 private void buatNotifikasi(){
    String metode = Metode.getSelectedItem().toString();
2
3     if (metode.equals("Suntik 1 Bulan")) {
        AlarmManager alarmManager = (AlarmManager)
        getSystemService(ALARM_SERVICE);
        Intent notificationIntent = new Intent(this,
        PenerimaKalender.class);
        PendingIntent broadcast =
        PendingIntent.getBroadcast(this, 100, notificationIntent,
        PendingIntent.FLAG_UPDATE_CURRENT);
        Calendar cal = Calendar.getInstance();
        cal.add(Calendar.MONTH, 1);
        alarmManager.setExact(AlarmManager.RTC_WAKEUP,
        cal.getTimeInMillis(), broadcast);
    } else (metode.equals("Suntik 3 Bulan")) {
4
        AlarmManager alarmManager = (AlarmManager)
        getSystemService(Context.ALARM_SERVICE);
        Intent notificationIntent = new Intent(this,
        PenerimaKalender.class);
        PendingIntent broadcast =
        PendingIntent.getBroadcast(this, 100, notificationIntent,
        PendingIntent.FLAG_UPDATE_CURRENT);
        Calendar cal = Calendar.getInstance();
        cal.add(Calendar.MONTH, 3);
        alarmManager.setExact(AlarmManager.RTC_WAKEUP,
        cal.getTimeInMillis(), broadcast);
5
6
7     }

```

Gambar 6.3 Kode Program *Method* buatNotifikasi

Berdasarkan proses pengujian unit yang telah dilakukan sebelumnya, maka *method* buatNotifikasi() akan menghasilkan *flowgraph* seperti yang ditunjukkan oleh Gambar 6.4.



Gambar 6.4 Flowgraph Method buatNotifikasi

Setelah diketahui *flowgraph* method `buatNotifikasi`, maka selanjutnya akan dihitung *cyclomatic complexity* menggunakan persamaan $V(G) = E - N + 2$. $V(G)$ merupakan jumlah kompleksitas siklomatis, E atau *edge* merupakan garis penghubung antar node, dan N merupakan jumlah node itu sendiri.

$$\begin{aligned}
 V(G) &= 7 - 7 + 2 \\
 &= 0 + 2 \\
 &= 2
 \end{aligned}$$

Berdasarkan hasil dari perhitungan kompleksitas siklomatis, maka didapatkan satu basis set jalur *independent*, yaitu:

Jalur 1: 1 – 2 – 3 – 4 – 7

Jalur 2: 1 – 2 – 5 – 6 – 7

Setelah mendapatkan urutan node dari jalur *independent* maka tahap selanjutnya adalah melakukan pengujian jalurnya. Pengujian terhadap tiap jalur tersebut ditunjukkan oleh Tabel 6.2.

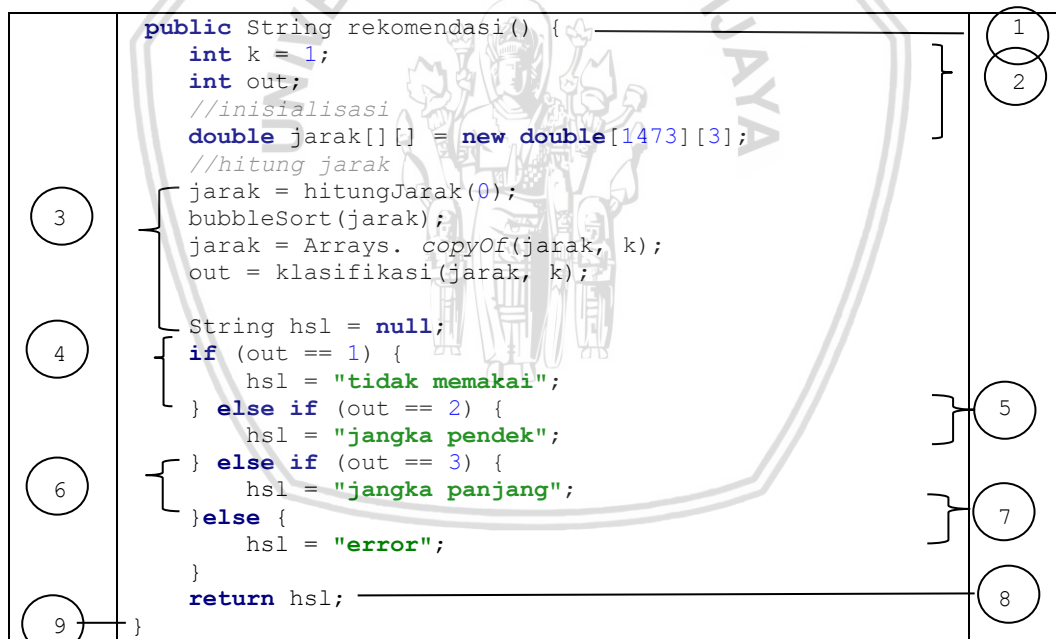
Tabel 6.2 Kasus Uji Pengujian Unit Method buatNotifikasi

Jalur	Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan
1	Membuat pemberitahuan suntik 1 bulan dengan memilih metode suntik yang telah disediakan pada form	<i>Method</i> <code>buatNotifikasi()</code> mengambil nilai masukkan, jika memilih suntik 1 bulan maka nilai tersebut akan di set pada <code>AlarmManager</code> untuk menampilkan	<i>Method</i> <code>buatNotifikasi</code> akan memberikan pemberitahuan suntuk 1 bulan

		pemberitahuan suntik 1 bulan	
2	Membuat pemberitahuan suntik 3 bulan dengan memilih metode suntik yang telah disediakan pada form	<i>Method</i> buatNotifikasi() mengambil nilai masukkan, jika memilih suntik 3 bulan maka nilai tersebut akan di set pada AlarmManager untuk menampilkan pemberitahuan suntik 3 bulan	<i>Method</i> buatNotifikasi akan memberikan pemberitahuan suntik 3 bulan

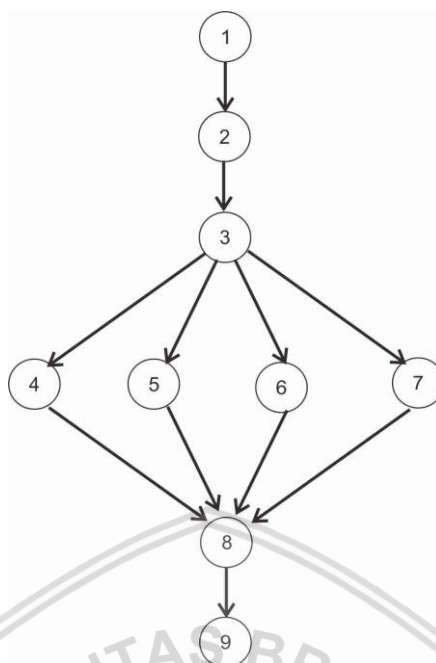
6.1.1.3 Pengujian *Method* rekomendasi()

Pengujian unit *method* rekomendasi() digunakan untuk menampilkan hasil dari perhitungan rekomendasi KB yang datanya yang sudah dimasukkan oleh akseptor. Kode program *method* rekomendasi() ditunjukkan oleh Gambar 6.3.



Gambar 6.5 Kode Program *Method* rekomendasi

Berdasarkan proses pengujian unit yang telah dilakukan sebelumnya, maka *method* rekomendasi() akan menghasilkan *flowgraph* seperti yang ditunjukkan oleh Gambar 6.4.



Gambar 6.6 Flowgraph Method rekomendasi

Setelah diketahui *flowgraph method* rekomendasi(), maka selanjutnya akan dihitung *cyclomatic complexity* menggunakan persamaan $V(G) = E - N + 2$. $V(G)$ merupakan jumlah kompleksitas siklomatis, E atau *edge* merupakan garis penghubung antar node, dan N merupakan jumlah node itu sendiri.

$$\begin{aligned}
 V(G) &= 11 - 9 + 2 \\
 &= 2 + 2 \\
 &= 4
 \end{aligned}$$

Berdasarkan hasil dari perhitungan kompleksitas siklomatis, maka didapatkan satu basis set jalur *independent*, yaitu:

Jalur 1 : 1 – 2 – 3 – 4 – 8 – 9

Jalur 2 : 1 – 2 – 3 – 5 – 8 – 9

Jalur 3 : 1 – 2 – 3 – 6 – 8 – 9

Jalur 4 : 1 – 2 – 3 – 7 – 8 – 9

Setelah mendapatkan urutan node dari jalur *independent* maka tahap selanjutnya adalah melakukan pengujian jalurnya. Pengujian terhadap tiap jalur tersebut ditunjukkan oleh Tabel 6.2.

Tabel 6.3 Kasus Uji Pengujian Unit Method rekomendasi

Jalur	Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan
1	Menampilkan hasil dari rekomendasi KB	<i>Method</i> tampilRekom() dapat menampilkan hasil dari perhitungan	<i>Method</i> tampilRekom() menampilkan hasil

		KB tidak memakai dengan	rekomendasi tidak memakai
2	Menampilkan hasil dari rekomendasi KB	<i>Method</i> tampilRekom() dapat menampilkan hasil dari perhitungan KB jangka pendek	<i>Method</i> tampilRekom() menampilkan hasil rekomendasi jangka pendek
3	Menampilkan hasil dari rekomendasi KB	<i>Method</i> tampilRekom() dapat menampilkan hasil dari perhitungan KB jangka panjang	<i>Method</i> tampilRekom() menampilkan hasil rekomendasi jangka panjang
4	Menampilkan hasil dari rekomendasi KB	Jika data yang dimasukkan tidak sesuai maka variable hsl akan mengembalikan nilai error	Sistem mengembalikan nilai error

6.1.2 Pengujian Validasi

Pengujian validasi berfungsi untuk mengetahui apakah sistem yang telah dibangun sudah sesuai dengan kebutuhan fungsional yang sudah di rancang pada tahap perancangan. Metode pengujian yang digunakan untuk melakukan pengujian validasi adalah metode pengujian *Black box*. Hal ini dikarenakan pengujian validasi lebih berfokus untuk mengamati hasil eksekusi melalui kasus uji. Pada pengujian validasi akan terbagi menjadi tiga bagian yaitu, pengujian validasi sisi pengguna, pengujian validasi sisi pemesan dan pengujian validasi sisi petugas.

6.1.2.1 Pengujian Validasi

(a) Kasus uji masuk ke sistem

Tabel 6.4 Kasus Uji Masuk ke Sistem

Nama Kasus Uji	Masuk ke Sistem
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat masuk kedalam sistem
Prosedur Pengujian	1. Memasukkan Alamat <i>email</i>

	2. Memasukkan Kata Sandi 3. Menekan tombol masuk
Hasil yang Diharapkan	Sistem dapat menyimpan data pengguna ke dalam <i>firebase database</i> dan menampilkan halaman beranda.

(b) Kasus uji daftar akun

Tabel 6.5 Kasus Uji Daftar Akun

Nama Kasus Uji	Daftar Akun
Tujuan Pengujian	Untuk membuktikan bahwa pengguna dapat mendaftarkan akun
Prosedur Pengujian	1. Memilih tombol daftar pada halaman masuk ke sistem 2. Memasukkan Alamat <i>email</i> 3. Memasukkan Kata Sandi 4. Menekan tombol daftar
Hasil yang Diharapkan	Sistem dapat menyimpan data pengguna ke dalam <i>firebase database</i> dan menampilkan halaman masuk ke sistem.

(c) Kasus uji melihat informasi KB

Tabel 6.6 Kasus uji Melihat Informasi KB

Nama Kasus Uji	Melihat Informasi KB
Tujuan Pengujian	Untuk membuktikan bahwa akseptor dapat melihat informasi KB
Prosedur Pengujian	1. Memasukkan email 2. Memasukkan password 3. Menekan tombol Login
Hasil yang Diharapkan	Pengguna dapat masuk ke dalam sistem dan akan menampilkan halaman menu utama

(d) Kasus uji membuat rekomendasi KB

Tabel 6.7 Kasus Uji Membuat Rekomendasi KB

Nama Kasus Uji	Membuat Rekomendasi KB
Tujuan Pengujian	Untuk membuktikan bahwa akseptor dapat membuat rekomendasi KB
Prosedur Pengujian	1. Memilih menu rekomendasi KB pada halaman beranda

	2. Mengisi form rekomendasi KB pada halaman membuat rekomendasi KB 3. Memilih tombol hitung
Hasil yang Diharapkan	Sistem akan menampilkan halaman membuat rekomendasi

(e) Kasus uji lihat rekomendasi KB

Tabel 6.8 Kasus Uji Lihat Rekomendasi KB

Nama Kasus Uji	Lihat Rekomendasi KB
Tujuan Pengujian	Untuk membuktikan bahwa akseptor dapat melihat rekomendasi KB
Prosedur Pengujian	1. Memilih menu rekomendasi KB pada halaman beranda 2. Mengisi form rekomendasi KB pada halaman membuat rekomendasi KB 3. Memilih tombol hitung 4. Melihat hasil rekomendasi KB
Hasil yang Diharapkan	Sistem akan menampilkan hasil dari rekomendasi KB

(f) Melihat Riwayat Kalender KB

Tabel 6.9 Kasus Uji Lihat Riwayat Kalender KB

Nama Kasus Uji	Lihat Riwayat Kalender KB
Tujuan Pengujian	Untuk membuktikan bahwa akseptor dapat melihat riwayat kalender KB
Prosedur Pengujian	1. Memilih menu kalender KB pada halaman beranda 2. Melihat riwayat kalender KB
Hasil yang Diharapkan	Sistem akan menampilkan hasil dari rekomendasi KB

(g) Kasus uji Buat Kalender KB

Tabel 6.10 Kasus Uji Buat Kalender KB

Nama Kasus Uji	Buat Kalender KB
Tujuan Pengujian	Untuk membuktikan bahwa akseptor dapat membuat kalender KB

Prosedur Pengujian	<ol style="list-style-type: none"> 1. Memilih menu kalender KB pada halaman beranda 2. Memilih tombol tambah riwayat pada halaman riwayat kalender KB 3. Memasukkan data sesuai dengan form yang telah tersedia pada halaman tambah waktu KB 4. Memilih tombol buat pemberitahuan
Hasil yang Diharapkan	Sistem dapat membuat kalender KB

(h) Kasus uji Menerima Pemberitahuan KB

Tabel 6.11 Kasus Uji Menerima Pemberitahuan KB

Nama Kasus Uji	Menerima Pemberitahuan KB
Tujuan Pengujian	Untuk membuktikan bahwa akseptor dapat menerima pemberitahuan KB
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Sistem menampilkan pemberitahuan KB jika sudah waktunya melakukan KB
Hasil yang Diharapkan	Sistem akan menerima pemberitahuan KB

(i) Kasus uji Lihat Kalender KB

Tabel 6.12 Kasus Uji Lihat Pemberitahuan KB

Nama Kasus Uji	Lihat Pemberitahuan KB
Tujuan Pengujian	Untuk membuktikan bahwa akseptor dapat melihat pemberitahuan KB
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Sistem menampilkan waktu KB jika sudah sesuai dengan tanggal yang dimasukkan dalam form buat kalender KB
Hasil yang Diharapkan	Sistem akan menampilkan pemberitahuan KB

(j) Kasus uji Mencari Rumah Sakit Terdekat

Tabel 6.13 Kasus Uji Mencari Rumah Sakit Terdekat

Nama Kasus Uji	Mencari Puskesmas Terdekat
Tujuan Pengujian	Untuk membuktikan bahwa akseptor dapat mencari puskesmas terdekat

Prosedur Pengujian	<ol style="list-style-type: none"> 1. Memilih menu Rumah Sakit pada halaman beranda 2. Memasukkan Kota atau tempat rumah sakit yang ingin dicari 3. Memilih tombol cari
Hasil yang Diharapkan	Sistem akan menampilkan posisi rumah sakit terdekat dari lokasi yang telah dimasukkan

(k) Kasus uji Melihat Profil

Tabel 6.14 Kasus Uji Melihat Profil

Nama Kasus Uji	Melihat Profil
Tujuan Pengujian	Untuk membuktikan bahwa sistem dapat menampilkan profil akseptor
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Memilih menu profil pada halaman beranda
Hasil yang Diharapkan	Sistem akan menampilkan halaman profil

(l) Kasus uji Mengubah Profil

Tabel 6.15 Kasus Uji Mengubah Profil

Nama Kasus Uji	Mengubah Profil
Tujuan Pengujian	Untuk membuktikan bahwa sistem dapat merubah profil akseptor
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Memilih menu profil pada halaman beranda 2. Memilih menu edit profil pada halaman profil 3. Memasukkan data sesuai dengan form yang telah disediakan pada halaman edit profil 4. Memilih tombol simpan
Hasil yang Diharapkan	Sistem akan menyimpan perubahan data yang telah dibuat di <i>firebase database</i>

(m) Kasus uji Keluar dari Sistem

Tabel 6.16 Kasus Uji Keluar dari Sistem


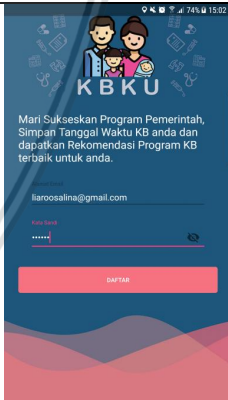
Nama Kasus Uji	Keluar dari Sistem
Tujuan Pengujian	Untuk membuktikan bahwa akseptor dapat keluar dari sistem

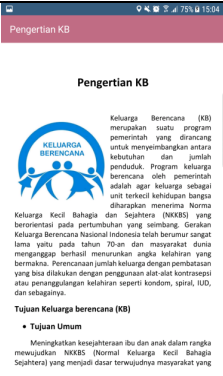

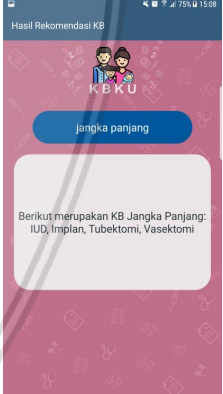

Prosedur Pengujian	<ol style="list-style-type: none"> Memilih menu profil pada halaman beranda Memilih menu keluar pada halaman profil
Hasil yang Diharapkan	Akseptor dapat keluar dari sistem

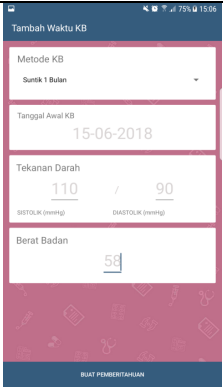


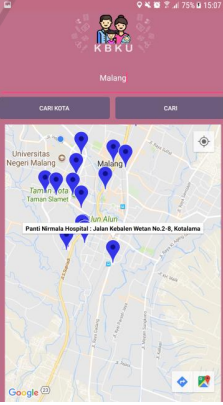
6.1.2.2 Hasil Pengujian Validasi

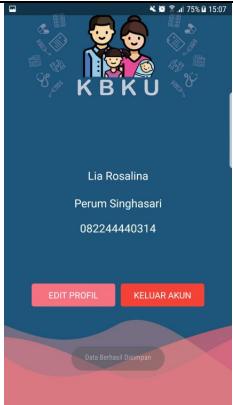
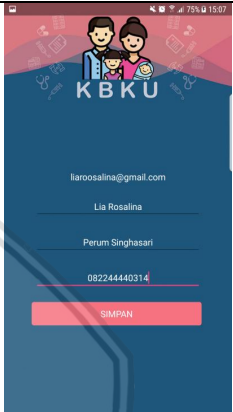
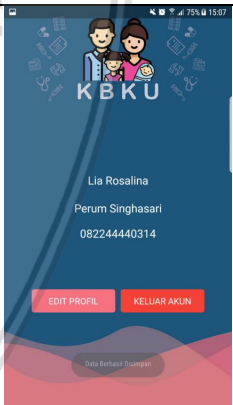
Tabel 6.17 merupakan hasil dari pengujian validasi yang telah dilakukan pada aplikasi KBku.

Tabel 6.17 hasil pengujian validasi sisi pengguna

No	Nama Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan	Screenshot Validasi	Status Validasi
1	Masuk ke Sistem	Pengguna dapat masuk ke dalam sistem dan menampilkan halaman beranda	Pengguna masuk ke dalam sistem dan menampilkan halaman beranda		Valid
2	Daftar Akun	Pengguna dapat mendaftarkan akun baru dan data disimpan di <i>firebase database</i> dan menampilkan halaman daftar akun	Sistem dapat membuat akun baru		Valid

3	Melihat Informasi KB	Sistem dapat menampilkan informasi KB pada halaman informasi KB	Sistem dapat menampilkan halaman informasi KB		Valid
4	Membuat Rekomendasi KB	Sistem dapat membuat rekomendasi KB pada halaman rekomendasi KB	Sistem dapat menampilkan halaman rekomendasi KB		Valid
5	Melihat Rekomendasi KB	Sistem dapat menampilkan halaman dari hasil rekomendasi KB	Sistem dapat menampilkan hasil rekomendasi KB		Valid
6	Melihat Riwayat Kalender KB	Sistem dapat menampilkan halaman riwayat Kalender KB	Sistem dapat menampilkan riwayat kalender KB		Valid

7	Membuat Kalender KB	Sistem dapat membuat kalender KB pada halaman buat kalender KB	Sistem dapat membuat kalender KB		Valid
8	Menerima Pemberitahuan KB	Sistem dapat menerima pemberitahuan KB	Sistem dapat menerima pemberitahuan KB		Valid
9	Melihat Kalender KB	Sistem dapat menampilkan halaman pemberitahuan kalender KB	Sistem dapat menampilkan halaman pemberitahuan kalender KB		Valid
10	Mencari Rumah Sakit Terdekat	Sistem dapat melakukan pencarian rumah sakit terdekat	Sistem dapat menampilkan rumah sakit terdekat		Valid

11	Melihat Profil	Sistem dapat menampilkan profil yang telah dibuat oleh akseptor	Sistem dapat menampilkan halaman profil		Valid
12	Mengubah Profil	Sistem dapat mengubah profil	Sistem dapat mengubah profil		Valid
13	Keluar Sistem	Akseptor dapat keluar dari sistem	Akseptor dapat keluar dari sistem		Valid

6.2 Pengujian Non-Fungsional

Pada penelitian ini akan dilakukan pengujian terhadap kebutuhan non-fungsional. Kebutuhan non-fungsional yang akan dilakukan pengujian adalah kebutuhan non-fungsional *usability* dan *compatibility*.

6.2.1 Pengujian *Usability*

Pengujian *usability* digunakan untuk menganalisis kebutuhan non-fungsional. Pengujian *usability* pada aplikasi KBku digunakan untuk melihat apakah aplikasi KBku sudah memenuhi ketentuan dari metode *usability*. Untuk pengujian pada metode *usability* sendiri terdapat dua metode pengujian, yaitu dengan menggunakan *task scenario* dan *USE Quisonnaire*. Pengujian pertama dengan menggunakan *task scenario*. Pengujian *task scenario* pada aplikasi KBku digunakan dengan menunjukan aplikasi KBku dan melihat bagaimana respon pengguna KB terhadap aplikasi KBku. Untuk pengujian *usability* melibatkan 8 orang responden yang terdiri dari 5 pengguna KB dan 3 bidan.

6.2.1.1 Pengujian *Usability* Menggunakan *Task Scenario*

Pengujian *task scenario* yang dilakukan pada pengujian *usability* ini dilakukan kepada 8 responden dengan memberikan 13 *task* yang harus dijalankan. Setelah pengguna selesai menjalankan aplikasi berdasarkan *task* yang telah dibuat maka akan dilakukan proses verifikasi apakah dari setiap *task* yang telah diberikan dapat dijalankan dengan baik atau gagal dilakukan. *Task* yang dilakukan dijelaskan pada Tabel 6.16.

Tabel 6.18 Pengujian *Usability* Menggunakan *Task Scenario*

No	Task	Task Description
1	Pengguna aplikasi KBku dapat masuk kedalam sistem	Pengguna dapat masuk ke dalam sistem dan melihat menu beranda pada aplikasi KBku
2	Pengguna aplikasi KBku dapat mendaftarkan akun baru	Pengguna dapat mendaftarkan akun jika belum memiliki akun KBku
3	Akseptor dapat melihat informasi KB	Akseptor dapat melihat informasi KB
4	Akseptor dapat membuat rekomendasi KB	Akseptor dapat membuat rekomendasi KB dengan memasukkan data yang telah tersedia pada form rekomendasi KB
5	Akseptor dapat melihat hasil rekomendasi KB	Akseptor dapat melihat hasil rekomendasi KB

6	Akseptor dapat melihat riwayat kalender KB	Akseptor dapat melihat riwayat kalender KB yang telah dibuat sebelumnya
7	Akseptor dapat membuat Kalender KB	Akseptor dapat membuat Kalender KB dengan memasukkan data sesuai dengan form yang terdapat pada halaman buat kalender KB
8	Akseptor dapat menerima pemberitahuan KB	Akseptor dapat menerima pemberitahuan KB
9	Akseptor dapat melihat pemberitahuan KB	Akseptor dapat melihat pemberitahuan waktu KB yang telah dibuat
10	Akseptor dapat mencari rumah sakit terdekat	Akseptor dapat mencari rumah sakit terdekat
11	Akseptor dapat melihat profil	Akseptor dapat melihat profil
12	Akseptor dapat merubah profil	Akseptor dapat merubah profil
13	Akseptor dapat keluar dari sistem	Akseptor dapat keluar dari sistem

Tabel 6.19 Hasil Pengujian *Task Scenario*

No	Pengguna KBku	Hasil Pengujian													Ket
		T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	
1	Luluk Setiyo Sejati	√	√	√	√	√	√	√	√	√	√	√	√	√	-
2	Nur Saidah	√	√	√	√	√	√	√	√	√	√	√	√	√	-
3	Farida Elvia	√	√	√	√	√	√	√	√	√	√	√	√	√	-
4	Sofia	√	√	√	√	√	√	√	√	√	√	√	√	√	-
5	Selvi	√	√	√	√	√	√	√	√	√	√	√	√	√	-
6	Tiara Alief, AMd.Keb	√	√	√	√	√	√	√	√	√	√	√	√	√	-
7	Siti Nurjanah, AMd.Keb	√	√	√	√	√	√	√	√	√	√	√	√	√	-
8	Dewi Khanzah, AMd.Keb	√	√	√	√	√	√	√	√	√	√	√	√	√	-
<i>Hasil</i>		100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	-

Setelah didapatkan hasil dari *task* yang telah dibuat, maka hasilnya akan dianalisis. Hasil dari analisis *task scenario* dijelaskan di Tabel 6.19. Pada tabel tersebut dijelaskan sukses atau gagalnya suatu *task*. Dan alasan kenapa *task* tersebut tidak bisa dijalankan. Pada Tabel pengujian *task scenario* pengguna aplikasi KBku terdapat simbol (√) yang artinya dapat dijalankan dan (X) yang berarti tidak dapat dijalankan.

6.2.1.2 Pengujian Usability Menggunakan USE Quissionaire

Use Quissionaire digunakan untuk menguji secara langsung aplikasi yang telah dikembangkan. Sebelum menyebar kuisioner, maka pengguna KB harus sudah mencoba aplikasi KBku terlebih dahulu. Kemudian kuisioner baru disebar kepada 8 responden yang sudah ditetapkan sebelumnya. Hasil dari kuisioner yang telah disebar terdapat pada Tabel 6.20.

Tabel 6.20 Pengujian Usability Menggunakan USE Quissionaire

USEFULLNESS		STS	TS	N	S	SS
1	Aplikasi KBku efektif untuk membantu melakukan program KB	-	-	-	2	6
2	Aplikasi KBku dapat membantu mempermudah mengetahui waktu jadwal kontrol KB	-	-	-	-	8
3	Aplikasi KBku dapat mempermudah penyimpanan riwayat kesehatan pada saat melakukan kontrol KB	-	-	-	-	8
4	Aplikasi KBku mampu memberikan rekomendasi jenis KB yang sesuai dengan kondisi pengguna KB	-	-	2	3	3
5	Aplikasi KBku mampu memberikan informasi rumah sakit terdekat	-	-	-	-	8
6	Aplikasi KBku dapat mempermudah memberikan informasi untuk melakukan program KB	-	-	-	-	8
EASE OF USE						
8	Aplikasi KBku mudah dimengerti	-	-	-	-	8
9	Untuk menggunakan aplikasi KBku tidak memerlukan banyak tahapan	-	-	-	-	8
10	Dalam menggunakan aplikasi KBku hanya memerlukan sedikit usaha	-	-	-	-	8
11	Saya dapat menggunakan aplikasi KBku tanpa petunjuk penggunaan	-	-	1	1	6
12	Aplikasi KBku berfungsi dengan konsisten ketika digunakan	-	-	-	-	8
13	Aplikasi KBku dapat diakses setiap saat	-	-	1	-	7




14	Saya bisa mengatasi kesalahan ketika menggunakan aplikasi KBku dengan cepat dan mudah	-	-	2	-	6
15	Aplikasi KBku akan disukai oleh wanita usia subur yang akan atau sudah menggunakan KB serta tenaga kesehatan terkait	-	-	-	-	8
EASY OF LEARNING						
16	Saya dapat mempelajari penggunaan aplikasi KBku dengan cepat	-	-	-	2	6
17	Saya dapat mengingat cara penggunaan aplikasi KBku dengan mudah	-	-	-	-	8
18	Saya dapat dengan mudah memahami cara penggunaan aplikasi KBku	-	-	-	-	8
19	Saya dapat dengan cepat dan terampil dalam menggunakan aplikasi KBku	-	-	1	2	5
SATISFACTION						
20	Saya merasa puas dengan aplikasi KBku	-	-	-	-	8
21	Saya akan merekomendasikan aplikasi KBku kepada orang lain	-	-	-	-	8
22	Sangat menyenangkan dalam menggunakan aplikasi KBku	-	-	2	-	6
23	Aplikasi KBku bekerja sesuai dengan keinginan saya	-	-	2	1	5
24	Saya nyaman menggunakan aplikasi KBku	-	-	-	-	8
25	Saya harus memiliki aplikasi ini	-	-	1	1	6

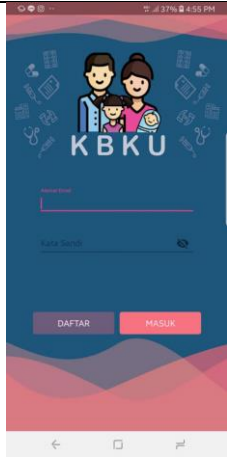
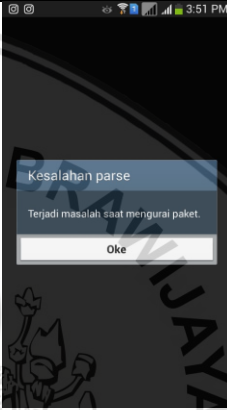
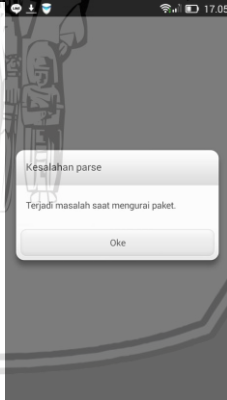
6.2.2 Pengujian *Compatibility*

Pengujian *compatibility* digunakan untuk mengetahui apakah aplikasi KBku dapat berjalan pada perangkat keras dan sistem operasi yang berbeda. Berikut merupakan hasil dari pengujian *compatibility* pada aplikasi KBku.

6.2.2.1 *Pengujian Compatibility*

Tabel 6.21 Hasil Pengujian *Compatibility*

No	Perangkat	Screenshot	Status Validasi
1	Xiaomi Redmi Note 3 <ul style="list-style-type: none"> - Ukuran Layar: 5.5 inches, 82.6 cm² - Resolusi layar: 1080 x 1920 pixels, 16:9 ratio (~403 ppi density) - Sistem Operasi: Android 5.0.2 (Lollipop) - API 21 		Valid
2	Samsung Galaxy S7 Edge <ul style="list-style-type: none"> - Ukuran Layar: 5.5 inches, 83.4 cm² - Resolusi Layar: 1440 x 2560 pixels, 16:9 ratio (~534 ppi density) - Sistem Operasi: Android 7.0 (Nougat) - API 24 		Valid
3	Sony Xperia XA Ultra Dual <ul style="list-style-type: none"> - Ukuran Layar: 6.0 inches, 99.2 cm² - Resolusi Layar: 1080 x 1920 pixels, 16:9 ratio (~367 ppi density) - Sistem Operasi: Android 7.0 (Nougat) - API 24 		Valid

4	<p>Samsung Galaxy S8</p> <ul style="list-style-type: none"> - Ukuran Layar: 5.8 inches, 84.8 cm² - Resolusi Layar: 1440 x 2960 pixels, 18.5:9 ratio (~570 ppi density) - Sistem Operasi: Android 8.0 (Oreo) - API 26 		Valid
5	<p>Samsung Galaxy Mega 5,8"</p> <ul style="list-style-type: none"> - Ukuran layar: 5.8 inches, 92.7 cm² - Resolusi Layar: 540 x 960 pixels, 16:9 ratio (~190 ppi density) - Sistem Operasi: Android 4.2.2 (Jelly Bean) - API 16 		Tidak Valid
6	<p>Lenovo S850</p> <ul style="list-style-type: none"> - Ukuran Layar: 5.0 inches, 68.9 cm² (~68.8% screen-to-body ratio) - Resolusi Layar: 720 x 1280 pixels, 16:9 ratio (~294 ppi density) - Sistem Operasi: Android 4.2 (Jelly Bean) - API 16 		Tidak Valid

6.2.2.2 Analisis Hasil Pengujian

Analisis hasil pengujian menjelaskan tentang hasil dari pengujian yang telah dilakukan pada aplikasi KBku pada perangkat android. Hasil analisis yang dilakukan meliputi pengujian unit, pengujian validasi, pengujian *usability*, dan pengujian *compatibility*.

6.2.3 Analisis Hasil Pengujian Unit

Analisis hasil dari pengujian unit yang telah dilakukan dengan cara menggunakan *white-box testing* dengan *basis path testing* serta melihat apakah fungsi dari kebutuhan fungsional yang ada telah diimplementasikan dalam unit

modul yang telah diuji memiliki kesesuaian dengan tahap perancangan yang telah dilakukan sebelumnya.

6.2.4 Analisis Hasil Pengujian Validasi

Analisis dari pengujian validasi dilakukan menggunakan cara perbandingan hasil uji dan daftar kebutuhan yang ada. Jika sudah sesuai dengan scenario yang ada pada sistem maka hasil dari pengujian tersebut *valid*. Namun jika didapati hasil yang tidak *valid* maka sistem belum memenuhi kebutuhan fungsionalnya. Untuk aplikasi KBku sendiri telah memenuhi kebutuhan fungsional yang sudah ditetapkan sebelumnya.

6.2.5 Analisis Hasil Pengujian Usability

Berikut merupakan analisis dari hasil pengujian *usability*. Untuk pengujian *usability* pada pembuatan aplikasi KBku menggunakan dua metode yaitu *task scenario* dan *USE Quissionaire*.

6.2.5.1 Analisis Hasil Pengujian Usability Menggunakan Task Scenario

Berdasarkan hasil dari pengujian *task scenario* yang dilakukan pada aplikasi KBku dengan yang diberikan pada 5 pengguna KB dan 3 bidan maka didapatkan hasil 100% pada seluruh *task* yang digunakan. Hasil rata-rata nilai pada pengujian menggunakan *task scenario* adalah sebagai berikut:

Tabel 6.22 Analisis Hasil Pengujian Task Scenario

No	Pengguna	Rata-rata
1	Pengguna KB	100%

6.2.5.2 Analisis Hasil Pengujian Usability Menggunakan USE Quissionaire

Pengujian dengan menggunakan *USE Quissionaire* ini memiliki tujuan untuk mengetahui sejauh mana mana tingkat kegunaan, kemudahan dan kepuasan pengguna dalam menggunakan aplikasi yang sedang dikembangkan. Hasil pengujian aplikasi KBku dengan menggunakan *USE Quissionaire* dapat dilihat pada Tabel

Tabel 6.23 Analisis Hasil Pengujian USE Quissionaire

USEFULLNESS		KELAYAKAN	HASIL AKHIR
1	Aplikasi KBku efektif untuk membantu melakukan program KB	38	95%
2	Aplikasi KBku dapat membantu mempermudah mengetahui waktu jadwal kontrol KB	40	100%

3	Aplikasi KBku dapat mempermudah penyimpanan riwayat kesehatan pada saat melakukan kontrol KB	40	100%
4	Aplikasi KBku mampu memberikan rekomendasi jenis KB yang sesuai dengan kondisi pengguna KB	33	82,5%
5	Aplikasi KBku mampu memberikan informasi rumah sakit terdekat	40	100%
6	Aplikasi KBku dapat mempermudah memberikan informasi untuk melakukan program KB	40	100%
7	Aplikasi KBku mudah dimengerti	40	100%
	Rata-rata		96.78%
EASE OF USE			
8	Untuk menggunakan aplikasi KBku tidak memerlukan banyak tahapan	40	100%
9	Dalam menggunakan aplikasi KBku hanya memerlukan sedikit usaha	40	100%
10	Saya dapat menggunakan aplikasi KBku tanpa petunjuk penggunaan	37	92,5%
11	Aplikasi KBku berfungsi dengan konsisten ketika digunakan	40	100%
12	Aplikasi KBku dapat diakses setiap saat	38	95%
13	Saya bisa mengatasi kesalahan ketika menggunakan aplikasi KBku dengan cepat dan mudah	36	90%
14	Aplikasi KBku akan disukai oleh wanita usia subur yang akan atau sudah menggunakan KB serta tenaga kesehatan terkait	40	100%
	Rata-rata		96.78%
EASE OF LEARNING			
15	Saya dapat mempelajari penggunaan aplikasi KBku dengan cepat	38	95%
16	Saya dapat mengingat cara penggunaan aplikasi KBku dengan mudah	40	100%
17	Saya dapat dengan mudah memahami cara penggunaan aplikasi KBku	40	100%

18	Saya dapat dengan cepat dan terampil dalam menggunakan aplikasi KBku	36	90%
	Rata-rata		96,25%
SATISFACTION			
19	Saya merasa puas dengan aplikasi KBku	40	100%
20	Saya akan merekomendasikan aplikasi KBku kepada orang lain	40	100%
21	Sangat menyenangkan dalam menggunakan aplikasi KBku	36	90%
22	Aplikasi KBku bekerja sesuai dengan keinginan saya	35	87,5%
23	Saya nyaman menggunakan aplikasi KBku	40	100%
24	Saya harus memiliki aplikasi ini	37	92,5%
	Rata-rata		95%
	Rata-rata Keseluruhan		96,20%

Kelayakan yang ditunjukkan pada kriteria *usefulness*, *ease of use*, *ease of learning* dan *satisfaction* yang dikembangkan memiliki nilai 96,20%. Menurut tabel skala Likert hasil nilai yang didapat menunjukkan bahwa hasil dari pengujian usability menggunakan kuesioner USE dapat dikatakan sangat baik.

6.2.6 Analisis Hasil Pengujian *Compatibility*

Analisis hasil dari pengujian *compatibility* didapatkan dengan memasang aplikasi KBku pada 6 perangkat android dengan versi yang berbeda. Dari hasil pemasangan aplikasi KBku di 6 perangkat android versi yang berbeda didapatkan hasil bahwa empat dari enam perangkat sukses menjalankan aplikasi KBku. Dan dua perangkat gagal menjalankan aplikasi KBku, ini dikarenakan sistem operasi yang digunakan memakai versi 4.2.2 yang dimana untuk dapat menggunakan aplikasi KBku harus menggunakan android versi 5 keatas. Berdasarkan hasil dari pengujian *compatibility* maka didapatkan hasil yang *valid*, karena aplikasi KBku dapat dipasang dan dijalankan pada android dengan versi sistem operasi versi 5 keatas.

BAB 7 PENUTUP

Pada bagian ini merupakan bagian akhir dari laporan penelitian. Bagian ini berisi kesimpulan yang akan menjawab rumusan masalah yang telah dipaparkan sebelumnya pada bab pendahuluan, serta saran dari penulis yang dapat digunakan untuk pengembangan aplikasi lebih lanjut.

7.1 Kesimpulan

Berdasarkan analisis yang dilakukan peneliti pada penelitian ini, maka dapat ditarik kesimpulan diantaranya:

1. Pada proses pembangunan aplikasi perangkat bergerak program KB berbasis android dengan menerapkan *user centered design*, dilakukan tahapan analisis kebutuhan untuk menggali kebutuhan pengguna KB, pada tahapan analisa kebutuhan ini terdapat identifikasi aktor, analisis kebutuhan fungsional, analisis kebutuhan non fungsional, *use case diagram*, dan *use case scenario*. Tahapan analisa kebutuhan ini menerapkan tahapan UCD pada bagian *specify the context of use*, dan *specify user and orgnixational requirements*. Pada tahapan analisa kebutuhan terjadi 2 iterasi. Setelah dilakukan tahapan analisa maka masuk kedalam tahapan perancangan yang didalamnya terdapat perancangan arsitektur sistem, perancangan basis data, perancangan *class sequence*, perancangan *class diagram*, dan perancangan antarmuka. Pada tahapan UCD, perancangan ini termasuk dalam tahapan *produce design solution*. Berdasarkan iterasi yang sudah didapatkan maka pada tahapan perancangan aplikasi KBku ini memiliki 2 tahapan iterasi. Kemudian untuk mendapatkan evaluasi dan iterasi, menggunakan proses UCD pada tahapan *evaluate design against user requirements*, yang dilakukan dengan cara wawancara langsung terhadap pengguna KB.
2. Pada proses pembangunan aplikasi perangkat bergerak program KB setelah dilakukan proses perancangan selesai dan tidak mendapatkan evaluasi dari pengguna KB, maka masuk kedalam tahapan implementasi sistem. Pada tahapan ini, terdiri dari spesifikasi sistem, batasan implemetasi, implementasi basis data, implementasi *class*, implementasi kode program, dan implementasi antarmuka. Pada tahapan implementasi terdapat beberapa fitur yang diimplementasikan, salah satunya adalah fitur pemberitahuan KB, pada fitur pemberitahuan KB penyimpanan datanya menggunakan *firebase*, hal ini dikarenakan *firebase* memiliki sifat *real time* sehingga data dapat secara otomatis disimpan dan disinkronkan. Kemudian tedapat fitur rekomendasi KB, pada fitur ini menggunakan basis data *sqlite* sebagai penyimpanan datanya, hal ini dikarenakan data yang digunakan bersifat statis dan tidak berubah.

3. Pada pembangunan aplikasi KBku terdapat dua pengujian yaitu pengujian fungsional dan pengujian non fungsional. Pengujian fungsional dilakukan dengan dua cara yaitu pengujian unit dan validasi. Hal ini dilakukan untuk melakukan validasi apakah *method* yang dibuat telah berjalan sesuai dengan fungsinya. Berdasarkan hasil dari pengujian unit dan validasi didapatkan hasil bahwa pengujian pada aplikasi KBku 100% *valid*, artinya bahwa semua *method* yang dimiliki pada aplikasi KBku sudah berjalan sebagaimana mestinya.
4. Pada tahapan pengujian ini terdiri dari pengujian non fungsional yang dilakukan kepada delapan persona, yaitu lima pengguna KB, dan tiga bidan. Pada tahapan pengujian non fungsional, aplikasi KBku menggunakan pengujian *usability* dengan menggunakan *task scenario* dan *use quissionaire*. Hasil dari *task scenario* adalah 100%, hal ini didapatkan dari wawancara langsung dengan pengguna KB. Kemudian pada pengujian *usability* dengan *USE Quissionaire* didapatkan seluruh rata-rata hasil sebesar 96,20%. Hal ini didapatkan dengan cara mendatangi langsung pengguna KB dan memberikan kuisisioner yang berisikan *usefulness*, *ease of use*, *ease of learning*, dan *satisfaction*. Dari hasil tersebut maka dapat disimpulkan bahwa aplikasi KBku memiliki tingkat kemudahan yang sangat baik.

7.2 Saran

Berdasarkan hasil penelitian yang dilakukan, penulis dapat menyampaikan saran untuk penelitian selanjutnya diantaranya sebagai berikut:

1. Ditambahkannya aplikasi untuk bidan yang terhubung pada aplikasi KBku agar dapat memonitoring langsung perkembangan pengguna KB.
2. Penambahan variable pada fitur rekomendasi Kb agar hasil rekomendasi KB semakin akurat.

DAFTAR PUSTAKA

- A. S., R. & Shalahuddin, M., 2011. *Modul Pembelajaran Rekayasa Perangkat Lunak*. Bandung: Modula.
- Abras, C., Krichmar, D. M. & Preece, J., 2004. *User-Centered Design*. p. 14.
- Affandi, B., Adriaansz, G., Gunardi, E. R. & Koesno, H., 2012. *Buku Panduan Praktis Pelayanan Kontrasepsi*. 3 ed. Jakarta: PT Bina Pustaka Sarwono Prawirohardjo.
- Amadyah, A., 2008. Rancangan Sistem Pameran Online Menggunakan Metode UCD (User Centered Learning). *SNTS AKPRIND*.
- android, d., 2018. *Developer Android*. [Online] Available at: <https://developer.android.com/guide/practices/compatibility> [Accessed 19 July 2018].
- Barnum, C. M., 2011. Usability Testing Essentials: Ready, Set....Test!. 54(3), pp. 336 - 337.
- Bertram, D., n.d. *Libraries and Curtular Resources - University of Calgary*. [Online] Available at: pages.cpsc.ucalgary.ca/~saul/wiki/.../topic-dane-likert.doc [Accessed 3 July 2018].
- Burnette, E., 2009. *Introducing Google's Mobile Development Platform*. 2nd, Edition ed. Raleigh, North Carolina Dallas, Texas: The Pragmatic Bookshelf.
- Devi, K. R., Sen, A. & K., H., 2012. A Working Framework for the User-Centered Design Approach and a Survey of the available Methods. *International Journal of Schientific and Research Publication*, 2(4).
- Doshi, P., Jain, P. & Shakwala, A., 2014. Location Based Services and Integration of Google Maps in Android. *International Journal Of Engineering And Computer Science ISSN:2319-7242*, 3(3), p. 6.
- Gerhana, Y. A., Atmadja, A. R., Zulfikar, B. W. & Ashanti, N., 2007. The implementation of K-nearest neighbor algorithm in case-based reasoning model for forming automatic answer identity and searching answer similarity of algorithm case.
- Gov, U., 2018. *Usability.gov*. [Online] Available at: <https://www.usability.gov/how-to-and-tools/methods/scenarios.html> [Accessed 19 July 2018].
- Gumelar, S., 2017. *PHP Indonesia Chapter Jakarta*. [Online] Available at: <https://phpid-jakarta.github.io/kulwap/2017/09/30/onesignal-push-notification-sasono-gumelar.html> [Accessed 18 05 2018].

Harrison, R., Flood, D. & Duce, D., 2013. Usability of Mobile Applications: Literature Review and Rationale for a New usability Model. *Journal of Interaction Science*, p. 16.

Jaya Kumar C, K., 2002. *Aplikasi 'E-Learning' Dalam Pengajaran Dan Pembelajaran Di Sekolah-Sekolah Malaysia: Cadangan Perlaksanaan Pada Senario Masa Kini*. Malaysia: Pasukan Projek Rintis Sekolah Bestari Bahagian Teknologi Pendidikan.

Jokela, T., Iivari, N., Matero, J. & Karukka, M., 2003. The Standard of User-Centered Design and the Standard Definition of Usability: Analyzing ISO 13407 against ISO 9241-11. p. 8.

Khalkhali, I. E., 2014. The use of DOKEOS e-learning platform in a Moroccan Business School. *International Conference on Multimedia Computing and Systems (ICMCS)*, pp. 633 - 638.

Khedkar, S. & Thube, S., 2017. Real Time Databases for Application. *International Research Journal of Engineering and Technology*, 04(06), pp. 2078 - 2082.

Kinanti, V. N., Yamin, M. & Aksara, L. F., 2016. Prototype Penyaring Asap Rokok Pada Smoking Area Menggunakan Pulse Width Modulation (PWM) dan Fuzzy Tsukamoto. *semanTIK*, Volume 2, p. 8.

Kreibich, J. A., 2010. *Using SQLite*. 1st ed. Sebastopol, California: O'Reilly Media, Inc..

Kumar, K., Akhi, K., Gunti, S. K. & Reddy, M. P., 2016. Implementing Smart Home Using Firebase. *International Journal of Research in Engineering and Applied Sciences (IJREAS)*, 6(10), p. 7.

LeRouge, C. & Wickramasighe, N., 2013. A Review of User-Centered Design for Diabetes-Related Consumer. *Journal of Diabetes Science and Technology*, p. 18.

Liu, Z., Hu, Y. & Cai, L., 2014. Research on software security and compatibility test for mobile application.

Lund, A. M., 2016. Measuring Usability with the USE Questionnaire.

Margono, 2004. *Metodologi Penelitian Pendidikan*. Jakarta: PT. Rineka Cipta.

Mayers, G. J., 2004. *The Art of Software Testing, Second Edition*. Canada: John Wiley & Sons, Inc., Hoboken, New Jersey.

Nielsen, J., 2000. *Why You Only Need to Test with 5 Users*. [Online] Available at: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/> [Accessed 6 June 2018].

Nugroho, C. G., Nugroho, D. & Fitriani, S. H., 2015. SISTEM PENDUKUNG KEPUTUSAN UNTUK PEMILIHAN METODE KONTRASEPSI PADA PASANGAN USIA SUBUR DENGAN ALGORITMA K-NEAREST NEIGHBOUR (KKN). *Jurnal Ilmiah Sinus*, p. 10.

Nugroho, C. G., Nugroho, D. & Fitriasih, S. H., 2015. SISTEM PENDUKUNG KEPUTUSAN UNTUK PEMILIHAN METODE KONTRASEPSI PADA PASANGAN USIA SUBUR DENGAN ALGORITMA K-NEAREST NEIGHBOUR (KKN). *Jurnal Ilmiah SINUS*, pp. 21-30.

Nurhadryani, Y., Sianturi, S. K., Hermadi, I. & Khotimah, K., 2011. Usability Testing to Enhance Mobile Application User Interface. *journal.ipb.ac.id*, Volume 2, pp. 83-93.

Nuryati, S. & Fitria, D., 2014. Jurnal ilmiah kesehatan Diagnosis. *PENGARUH FAKTOR INTERNAL DAN FAKTOR EKSTERNAL TERHADAP PEMILIHAN ALAT KONTRASEPSI PADA AKSEPTOR BARU DI KABUPATEN BOGOR*, Volume 5, p. 632.

Pratama, A. R., 2016. *CODEPOLITAN*. [Online] Available at: <https://www.codepolitan.com/belajar-uml-sequence-diagram-57fdb1a5ba777-17044> [Accessed 9 July 2018].

Preece, J., 1994. *Human - Computer Interaction*. Addison: Wesley.

Presman, R. S., 1997. *Software Enigneering*. 4 ed. s.l.: Mc. Graw Hill.

Pressman, R. S., 2010. *Software Engineering A Practitioner's Approach*. 7th ed. s.l.:The McGraw-Hill Companies.

Pusuluri, N. R., 2006. *Software Testing Concepts And Tools*. s.l.:Dreamtech Press.

Pusuluri, N. R., 2006. Software Testing Concepts And Tools.. In: N. R. Pusuluri, ed. Delhi: s.l.:Dreamtech Press..

Rahadi, D. R., 2014. Pengukuran Usability Menggunakan USE Questionnaire Pada Aplikasi Android. *Journal Sistem Informasi (JSI)*, Volume 6, pp. 661-671.

Rianto Rahadi, D., 2014. Pengukuran Usability Sistem Menggunakan Use Questionnaire Pada Aplikasi Android. *Jurnal Sistem Informasi (JSI)*, 6(1), pp. 661-671.

Rubin, J. & Chrisnell, D., 2008. *Handbook Of Usability Testing*. 2nd ed. Indianapolis: Wiley Publishing.Inc.

Setia, L. D., 2012. Evaluasi Usability Untuk Mengetahui Akseptabilitas Aplikasi Berbasis Web. p. 11.

Soekartawi, 2002. *Prospek Pembelajaran Melalui Internet*. Jakarta, UT-Pustekkom dan IPTPI.

Soleha, S., 2016. STUDI TENTANG DAMPAK PROGRAM KELUARGA BERENCANA DI. *eJournal Ilmu Pemerintahan*, Volume 4, p. 14.

STAPIĆ, Z., 2013. *A PROPOSAL OF AN ONTOLOGY-BASED METHODOLOGICAL FRAMEWORK FOR MULTI-PLATFORM MOBILE APPLICATIONS DEVELOPMENT*. Alcalá de Henares (Madrid): University of Alcalá Computer Science Department, Postgraduate School Doctoral program "Information and Knowledge Engineering".

Statista, 2017. Statista. [Online]
Available at: <https://www.statista.com/topics/1002/mobile-app-usage/>
[Accessed 18 July 2018].

Statista, 2017. The Statistic Portal. [Online]
Available at: <https://www.statista.com/>
[Accessed 2018].

ST, H., 2015. *Cepat Mahir MySQL dan SQLite*. s.l.:Elex Media Komputindo.

Triadi, D., 2013. *Bedah Tuntas Fitur Android*. 1 ed. Yogyakarta: Jogja Great! Publisher.

Trisanti, I. & Nasriyah, 2016. Hubungan Dukungan Suami dalam Pemilihan Metode Kontrasepsi Jangka Panjang (MKJP). *The 4 th Univesity Research Coloquium*.

usability.gov, 2018. usability.gov. [Online]
Available at: <https://www.usability.gov/how-to-and-tools/methods/scenarios.html>
[Accessed 8 July 2018].

Widhiarso, W., J. & S., 2007. Metode UCD (User Centered Design) Untuk Rancangan Kios Informasi Studi Kasus : Rumah Sakit Bersalin XYZ. *STMIK MDP Palembang*.

Wilopo, A. S., 1997. Arah Dan Implementasi Kebijaksanaan Program Keluarga Berencana di Indonesia. p. 15.